

# Week 8: R

Griffin Dietz

CS 197 HCI Section

# Administrivia

Draft paper assignment: 2-phase submission

MTurk Sandbox clarification and 3 stage deployment process

R! (I will circulate to talk to teams about evaluation plans)

# Packages and Libraries

`install.packages("dplyr")` to install the first time you use a package, e.g., dplyr

`library("dplyr")` to load every time you use that package

For this tutorial, you will need tidyverse, which is a wrapper around a lot of packages useful for statistical analysis

# Assignment operators: <- vs =

<- is an assignment operator in R

= is both an assignment operator and is used to specify function arguments

*Example*

```
plot <- ggplot(data=data, mapping=aes(x = age_group, y = correct)) +  
  geom_line()
```

vs.

```
plot = ggplot(data=data, mapping=aes(x = age_group, y = correct)) +  
  geom_line()
```

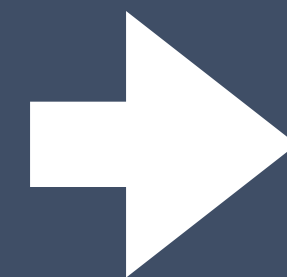
For clarity of code: use <-

# Tidy Data

One observation per row

Why? Consider: `t.test(cases~year)`

State	1997	2002	2007
AL	101	142	165
AK	62	87	99
AZ	214	233	249
AR	31	39	52



State	Year	Cases
AL	1997	101
AL	2002	142
AL	2007	165
AK	1997	62
AK	2002	87

# Piping in R

String together nested functions with `%>%` (in `dplyr` package). As operations get more complex, piping helps make code more readable

`round(sum(numbers), 2)` is the same as...

```
numbers %>%  
  sum() %>%  
  round(2)    #piping passes in prior computation as first argument
```

Helpful tip: `command-shift-m` types a `%>%` in RStudio (`control-shift-m` on Windows)

<https://github.com/gdietz44/R-tutorial>