**Studying for Test 2**

**Read over the Learning R, introduction to ggplot and introduction to dplyr (all posted on Blackboard at R Resources link). Go through powerpoints from class, and other readings. Review class scripts… The test will include:**

General questions about R and terminology, as on Test 1

Data Visualizations:

* Know the different main types of graphs (5NG). scatterplot(geom\_point), linegraph(geom\_line), bargraph(geom\_bar/geom\_col), histogram(geom\_histogram) and boxplot(geom\_boxplot).
* Know what can go wrong with graphs (disappearing baseline, dimensions in data not matching dimensions in the graph, etc.) Be prepared to calculate the Lie Factor
* Know what Tufte considers to be characteristics of good graphs (high data density, small multiples, etc.), as well as other opinions (e.g., on ChartJunk)
* Know what it means to ‘map a variable to an aesthetic’ and represent with a ‘geom’. Be able to write code for simple graphs.
* Know how to do the following coding tasks:
* Create a scatterplot with ggplot, using colored points, determined by another variable

ggplot(data = HI,

mapping = aes(x = score, y=gdpPerCapita, color=gdpPerCapita)) +

geom\_point()

* Create a bar graph, with bars filled with color, determined by another variable

ggplot(data = vores\_nona,

mapping = aes( x = vore, fill = conservation)) +

geom\_bar(position = 'fill')

* Create a histogram or density plot for one quantitative variable.

ggplot(data = HI,

mapping = aes(x = score)) +

geom\_histogram()

* Create a density plot, using facet\_grid or facet\_wrap, to introduce a second variable
* Use labs() to add a title, or a label for axes

msleep %>% filter(!is.na(vore)) %>%

ggplot(mapping = aes(x = sleep\_total, fill = vore)) +

geom\_density() +

labs( title = 'Total Hours of Sleep by "vore"', x = 'Total Hours of Sleep') +

facet\_wrap(~ vore, ncol = 1)

Data Transformation and Summary:

* Know what Tidy Data is

Tidy Data is **a way of structuring datasets to facilitate analysis**. ... In tidy data: Each variable must have its own column. Each observation must have its own row. Each type of observational unit forms a table.

* Know these data verbs in dplyr and what each is used for:

filter, mutate, group\_by, summarise, arrange, select, sample\_n

* Know how to do the following coding tasks:
* Finding mean, sd, median, etc., for a vector, using base package

mean(survey$Earnings)

sd(survey$Earnings)

median(survey$Earnings)

* Finding mean, sd, median, etc., for a vector, using dplyr (with summarise)

mpg %>% summarise(meanhwy = mean(hwy),

medhwy = median(hwy),

maxhwy = max(hwy),

minhwy = min(hwy),

sdhwy = sd(hwy),

sampsize = n())

* Know when **na.rm= TRUE** is needed, and where it goes (mainly, functions like mean())

???

* Use dplyr to find mean, sd, median, etc., by groups (using group\_by).

mpg %>% group\_by(class, drv) %>%

summarise(meanhwy = mean(hwy),

medhwy = median(hwy),

maxhwy = max(hwy),

minhwy = min(hwy),

sampsize = n())

* Filter data values and create a new (filtered) data frame.

portland\_flights <- flights %>%

filter(dest == "PDX")

* Filter out missing values for a particular variable from a data frame, using **!is.na().**

mpgnomiss <- mpg %>% filter(!is.na(hwy))

* Arrange a data frame in ascending or descending order

mpgarr <- mpg %>% arrange(hwy)

mpgarr <- mpg %>% arrange(desc(hwy))

* Print the first or last few observations of a data frame

… %>% head(6)

…%>%tail(6)

* Create a new variable, and add it to the current data frame (mutate)

mpg2 <- mpg %>% mutate(avemile = (cty + hwy)/2)