Lab 3 R Instructions **Ehsan Karim** *email :* `ehsan at stat dot ubc dot ca` (subj. line: 241/251)

#	R commands	Description of the command
1	`data()`	# look for R built-in data sets, see that there is a dataset **USArrests** for Violent Crime Rates by US State
2	`edit(USArrests)`	# take a look at the data, and close it

3	`sink("z:\\usa1.txt")`
	`USArrests`
	`sink()`
		# generate usa1.txt by exporting USArrests dataset in it (method 1 of export)

4	`write.table(USArrests, file = "z:\\usa2.txt")`	# generate usa2.txt by exporting USArrests dataset in it (method 2 of export)
5	`read.table("z:\\usa2.txt", header= T)`	# import external dataset usa2.txt in R, in which we have column names
6	`write.table(USArrests, file = "z:\\usa3.txt", sep = "\t")`	# generate usa3.txt by exporting USArrests dataset in it (method 2 of export), but this time with more control on how to separate the data elements. Open usa2.txt and usa3.txt to understand the difference clearly
7	`read.table("z:\\usa3.txt", header= T, sep = "\t")` `arrest = read.table("z:\\usa3.txt", header= T)`	# importing just like before, this time saving as an object in R
8	`dimnames(arrest)`	# check row and column names of the dataset
9	`attach(arrest)`	# Attach the dataset so that you can call each column names as an independent vector
10	`rank.murder = rank(Murder)` `rank.murder`	# find the rank of Murder.
11	`sum(rank.murder == 44)` `sum(rank.murder == 45)`	# Notice that there are fractions in rank. For example, Alabama [first row] and Tennessee [42nd row] has same Murder variable value = 13.2, thats why R makes an average of them, that is, assigns 44.5 in both, no 44 or 45.

12	`which(rank.murder == 44)`
	`which(rank.murder == 45)`
	`which(rank.murder == 44.5)`
		# We try to find value with rank 44 or 45, but there are none, instead, rank 44.5 is assigned with 1st and 42nd original Murder value.

13	`sort(rank.murder,index=T)`	# This will report two sets of numbers, the first ($x) is the ranks, and in order, and the second set ($ix) with the corresponding element position.
14	`rank.arrest = apply(arrest, MARGIN = 2, FUN = rank)` `rank.arrest`	# we calculate the rank of each variable via the apply(.) function, which can used to columns (2) of a data table.
15	`state.rank = apply(rank.arrest, MARGIN = 1, FUN = mean)` `state.rank`	# we calculate the mean of table via the apply(.) function, which can used to rows (1) of a data table.
16	`detach(arrest)`	# detach data file
17	`which(state.rank == min(state.rank))`	# finds where the crime index is minimum among all other states. This is the simple version of a nonparametric method by using the mean rank sum of the variables

18	`length(Murder)`
	`sample(c(0,1), size= 50, replace = T, prob = NULL)`
	`sample(c(0,1), size= 50, replace = T, prob = NULL) -> Party`
		# we generate a sample with 0,1 with size equal to size of the one Variable in the dataset (50). Name it. Say
		0 = party0,
		1 = party1

19	`rank.arrest = as.data.frame(rank.arrest)`	# treating rank.arrest as data frame
20	`dimnames(rank.arrest)` `rank.arrest$Murder`	# check variable names and use. Unless its a data frame, we cannot use $

21	`par(mfrow=c(2,2))`
	`boxplot(rank.arrest$Murder~Party)`
	`boxplot(rank.arrest$Assault~Party)`
	`boxplot(rank.arrest$Rape~Party)`
	`boxplot(rank.arrest$UrbanPop~Party)`
		# use boxplot for graphical presentation to find whether the arrest index varies depending on which party is in government