MEMO/ Tricks / Lessons learned in the STATS202A class

BHARGAV PARSI (804945591)

R

- Getting help
  1) ?mean -> get help of the mean function(or any function)
- Using Packages
  1) Install.packages('dplyr') -> used to install different packages
- Getwd() -> Find the current working directory
- Creating vectors
  1) C(2, 4, 6) -> 2 4 6
  2) 2 : 6 -> 2 3 4 5 6
  3) rep(1 : 2, times = 3) -> 1 2 1 2 1 2
  4) Selecting vector elements(in R indexing starts from 1)
     a. X[4] -> 4th element
     b. X[-4] -> All but the 4th element
     c. X[c(1,5)] -> Elements one to five
- Matrices
  1) M <- matrix(x, nrow = 3, ncol = 3)
  2) M[2,] – select a row
  3) M[,1] – select a column
  4) M[2,3] – select an element
  5) t(m) – Transpose

6) m %*% n -> matrix multiplication
7) solve(m,n) -> find x in: m * x = n

- Data Frames
  1) df <- data.frame(x = 1:3, y = c(‘a’,’b’,’c’))
  2) head(df) – see the first 6 rows
  3) nrow, ncol, dim(df) – number of rows and columns
  4) cbind, rbind -> bind rows and columns
  5) We can use the dataframe same as a matrix
- Statistics
  1) lm(y ~ x, data = df) -> Linear Model
  2) glm(y ~ x, data = df) -> generalized linear model
  3) summary -> More detailed information about a model.

RStudio

- Package Writing
  1) File > New Project > New Directory > R Package
     Turn project into package
  2) Enable roxygen documentation with
     Tools > Project Options > Buildtools
- Few Important Shortcuts
  1) Show packages -> Ctrl + 7
  2) Show Plots -> Ctrl + 6
  3) Source a File -> Ctrl + alt + G
  4) Run current Line/selection -> Ctrl + Enter
  5) Source current file -> Ctrl + Shift + S
  6) Source with Echo -> Ctrl + Shift + Enter
  7) Extract Function -> Ctrl + Alt + X
  8) Extract variable -> Ctrl + Alt + V
  9) Reindent Lines -> Ctrl + I
  10) Build and Reload -> Ctrl + Shift + B

11) Load All -> Ctrl + Shift + L
12) Document Package -> Ctrl + Shift + D

RCPP

- Always add #include <Rcpp.h> and using namespace Rcpp;
- Creating vectors
  1) NumericVector xx(10); -> array of 0 with size 10
  2) NumericVector xx(10, 2); -> default value is 2 instead of 0
  3) NumericVector xx(y.begin(), y.end()); -> Range constructor
- Using Matrices
  1) NumericMatrix xx(4,5); -> 4*5 matrix filled with 0
- Inline C++ compile in R:
  1) Use cppFunction("Define function in CPP")
  2) Use evalCpp("std::numeric_limits<double>::max()") for evaluating c++ expressions
- RcppArmadillo is another version of Rcpp whose syntax is much better to use.
- // [[Rcpp::depends(RcppArmadillo)]] -> always use this at the beginning of the file
- // [[Rcpp::export()]] -. Always use this at the beginning of every function
- **Some useful functions in Armadillo**
  1) Test <- rbind(m1, m2) → row wise binding of matrices
  2) cbind → columnwise binding
  3) rep(v, n) → repeat a vector for n times
  4) determimant(matrix) → finds determinant of matrix

R Parallel

- This technique allows us to execute r codes much faster.
- 2 packages must be installed first. Parallel and doparallel.
- Makecluster Is used to allocate the cores.
- Registerdoparallel → we need to register for parallel backend.
- Foreach will allow us to run loops in parallel.
- Cluster should be closed at the end of the execution.

Python

- Mind the white space. Compilation error with wrong spacing is very common.
- Reading from text documents is very easy in python. We just need to open the file and read line by line. After that, we can tokenize with respect to split(“ ”) method.
- Python list is one of the best data structures when compared with the arrays or lists in other languages. It can contain any number of integers, floating and different types can be present in a single list which makes our life very easy sometimes.
- ‘+’ can be used for concatenation. Other useful functions are sort() and insert()
- One dimensional fixed length, immutable sequence of python objects is a tuple which is another useful data structure. Usage can be found in swapping variables.
- Slicing can be done in lists which returns a specific portion of the list. For ex: list1[start:stop], list1[start:stop:step]
- Dict is a hashmap which provides us with o(1) search time. It maintains key→ value pair and if key does not exist it give a keyvalue error.

- Set is an unordered collection of unique elements. All ops in set theory can be performed on them.
- **Numpy:**
  1) Core library for scientific computing
  2) A.shape → Array dimensions
  3) Np.multiply → element wise multiplication
  4) Np.dot → inner product
  5) N.cumsum → cumulative sum
  6) Np.copy → copy of array
  7) Slicing → a[0:2], select items at index 0 and 1
  8) Np.transpose → transpose of array
  9) Np.resize → returns new array with new shape
  10) Np.column_stack → stack columnwise arrays
- **Scikit-Learn**
  1) It implements range of machine learning, preprocessing, cross validation and visualization algorithms using a united interface.
  2) Train_test_split → can be used to split data set to training and testing set.
  3) Preprocessing can be done by Standardization(StandardScaler()), Normalization(Normalizer())
  4) Linear Regression → from sklearn.linear_model import LinearRegression
  5) Support Vector Machine → SVC
  6) PCA can be implemented as PCA
  7) Fit → can be used to fit model to data

SQL

- SELECT columns_name FROM table_name → select data from table

- SELECT * FROM table_name → select all columns
- WHERE -> enables us to select data with certain properties from the data
- IN → may be used if you know the exact value you want to return for at least one of the columns
- ORDER BY → can be used to sort the output according to ASC(ascend) or DESC(descend)
- GROUP BY → commonly used with aggregate functions like SUM and return the aggregate of the particular group of columns values
- HAVING → similar to WHERE but used against Aggregate functions
- AVG, COUNT, MAX, MIN, SUM → Some aggragate functions
- INNER JOIN → returns all rows fromm both tables where there is a match
- LEFT JOIN → returns all the rows from the left table even if there are no matches in the second table
- RIGHT JOIN → similar to LEFT JOIN but it is done on the right table
- UNION → combine results from 2 SQL queries
- LIKE → to query rows in a table using regex

UNIX/LINUX

- Ls → directory listing
- Ls -al → formatted listing with hidden files
- Cd dir → change directory to dir
- Pwd → present working directory
- Mkdir → create a directory
- Rm → remove
- Cp file1 file2 → copy file
- Mv file1 file2 → move file

- Cat > file → std in to file
- More file → output the file std out
- Tail → output last 10 lines
- Chmod → change permissions of the file
- Grep → search for a pattern
- Pipe (|) symbol is used to pipe out the output of 1 command to input of other
- Sort → for sorting contents of list
- Who → list of currently logged in users
- .. → parent directory
- ~ → home directory
- Wc → count number of words
- < symbol is used to redirect standard input from a file

GITHUB

- **Create**
  1) Git init
  2) Git add
- **Show**
  1) Git status → files changed in working directory
  2) Git diff → changes made to tracked files
  3) Git log → history of changes
  4) Git blame → who changed what and when in file
- **Revert**
  1) Git reset → return to last committed state (permanent)
  2) Git revert HEAD → revert to last commit.
- **Update**
  1) Git fetch → fetches the latest changes
  2) Git pull → pull latest change
- **Publish**

1) Git commit -a → commit all local changes
2) Git push → push changes to origin

Hoffman

- Most powerful cluster in UC system
- To connect to Hoffman, use ssh login-id@hoffman2.idre.ucla.edu
- In windows we can use nomachine or putty
- Login Node → used for managing files
- Compute Node → obtained after the qrsh command. Jobs are executed here.
- 20 GB memory given. Temp storage of 2TB is available in '/u/scratch' for 7 days.
- Never use more memory, else job will be terminated
- Module can be used to load different applications such as R, Python
- Example command -> qrsh -l h_rt = 8:00:00,H_data=4G

Tensorflow

- It is an open source software library for numerical computation using data flow graphs.
- Installing → pip install tensorflow
- Tensor → generalization of vectors and matrices to high dimensions
- Placeholder → Used when we need to send data outside
- Session → used to send data into placeholder
- Cross-entropy → inefficiency of predictions for describing truth.
- **Main Classes**
  1) Tf.Graph()
  2) Tf.Operation()
  3) Tf.Tensor()

4) Tf.Session()

- **Tensorflow Optimizers**
  1) GradientDescentOptimizer
  2) AdadeltaOptimizer
  3) AdagradOptimizer
  4) MomentumOptimizer
  5) AdamOptimizer
  6) FtrlOptimizer
  7) RMSPropOptimizer
- **Reduction**
  1) Reduce_sum
  2) Reduce_prod
  3) Reduce_min
  4) Reduce_max
- **Activation Function**
  Relu, relu6, sigmoid, sigmoid_cross_entropy_with_logits, softmax, log_softmax
- Fit(X,y, monitor = None, logdir = None)
  X: tensor of shape [n_Samples, n_features]
  Y: vector of matrix [n_samples]
- Predict(X, axis = 1, batch_size = None)
  Axis : which axis to argmax for classification
  Use 2 for sequence predictions
  Batch_size: use this to split it into mini batches

SAS

- **Program Structure:**
  1) Data Step →To create a dataset that becomes the source of data analysis.
  2) Proc Step → This step analyses the data from above.

3) Output Step → shows the result of the analysis done in the proc step.

- Mean
  PROC MEANS DATA = dataset;
  Class Variables;
  VAR Variables;
- Standard deviation
  Proc means data = dataset STD;
- Correlation Analysis
  PROC CORR DATA = dataset options;
  VAR Variable;
- Linear Regression
  PROC REG DATA = dataset;
  MODEL variable1 = variable2;
- General Linear model
  Proc GLM DATA = dataset;
  MODEL model;
  RUN;
- Histogram:
  PROC UNIVARIATE DATA = dataset;
  HISTOGRAM variables;
  RUN;
- Bar Charts:
  PROC SGPLOT DATA = Dataset;
  VBAR variables;
  RUN;
- BOX PLOTS:
  PROC SGPLOT DATA = DATASET;

```
VBOX Variable/category = VARIABLE
RUN;
```