MONASH University
Information Technology

ASSESSMENT COVER SHEET

Student ID number	28525795
Given Name	Tianyue
Family name	Ma

Unit Name and Code:	**Advanced topic for information technology FIT5211**		
Campus:	**Caufield**		
Assignment Title:	Deployment of Dataset on MongoDB and R based simple analysis of Data		
Name of Lecturer:	Ariel Liebman		
Name of Tutor:	Mohamed Darweish		
Tutorial Day and Time:	**Thursday 2.00 pm – 4.00 pm**		
Phone Number:	**0423652806**		
Email Address:	**Tmaa0001@student.monash.edu**		
Has any part of this assignment been previously submitted as part of another unit/course? ☐ Yes ☒ No			
Due Date:	**09/13/2017**	Date Submitted:	09/13/2017

All work must be submitted by the due date. If an extension of work is granted this must be specified with the signature of the lecturer/tutor.

Extension granted until (date)_____ Signature of lecturer/tutor _____

Please note that it is your responsibility to retain copies of your assessments.

Intentional plagiarism or collusion amounts to cheating under Part 7 of the Monash University (Council) Regulations

Plagiarism: Plagiarism means taking and using another person's ideas or manner of expressing them and passing them off as one's own. For example, by failing to give appropriate acknowledgement. The material used can be from any source (staff, students or the internet, published and unpublished works).

Collusion: Collusion means unauthorised collaboration with another person on assessable written, oral or practical work and includes paying another person to complete all or part of the work.

Where there are reasonable grounds for believing that intentional plagiarism or collusion has occurred, this will be reported to the Associate Dean (Education) or delegate, who may disallow the work concerned by prohibiting assessment or refer the matter to the Faculty Discipline Panel for a hearing.

Student Statement:

- I have read the university's Student Academic Integrity Policy and Procedures.
- I understand the consequences of engaging in plagiarism and collusion as described in Part 7 of the Monash University (Council) Regulations http://adm.monash.edu/legal/legislation/statutes
- have taken proper care to safeguard this work and made all reasonable efforts to ensure it could not be copied.
- No part of this assignment has been previously submitted as part of another unit/course.
- I acknowledge and agree that the assessor of this assignment may for the purposes of assessment, reproduce the assignment and:
  - i. provide to another member of faculty and any external marker; and/or
  - ii. submit it to a text matching software; and/or
  - iii. submit it to a text matching software which may then retain a copy of the assignment on its database for the purpose of future plagiarism checking.
- I certify that I have not plagiarised the work of others or participated in unauthorised collaboration when preparing this assignment.

Signature .....Tianyue Ma..... Date……09/13/2017……

* delete (iii) if not applicable

The information on this form is collected for the primary purpose of assessing your assignment and ensuring the academic integrity requirements of the University are met. *Other purposes of collection include recording your plagiarism and collusion declaration, attending to course and administrative matters and statistical analyses. If you choose not to complete all the questions on this form it may not be possible for Monash University to assess your assignment. You have a right to access personal information that Monash University holds about you, subject to any exceptions in relevant legislation. If you wish to seek access to your personal information or inquire about the handling of your personal information, please contact the University Privacy Officer: privacyofficer@adm.monash.edu.au*

FIT5141 Assignment 2

The Deployment of Dataset on MongoDB and R based simple analysis of Data

Name: Tianyue Ma
Student number:28525795
Tutor's name: Mohamed Darweish
Tutorial time: Wednesday 2.00 pm – 4.00 pm

Table of Contents

About document

This document reports the implementation of installation and configuration of the data storage and simple analysis of energy consumption data. In the process of data analysis, the data is processed through several steps, such as data imports, data wrangling, and so on.

1. MongoDB server set up

Between two choices (set up own MongoDB server or Use university server), I set up my own MongoDB server. After downloading and installing the latest edition of MongoDB, there are several steps to set up the server:

Firstly, a data directory was created to store data by typing command “md \data\db” in a command prompt. Then we should type the command “C:\MongoDB\bin\mongod.exe” to start MongoDB and type the command “C:\MongoDB\bin\mongo.exe” in a new command prompt to connect with MongoDB.

```
C:\MongoDB\bin>mongod.exe
2017-09-13T04:02:57.862+1000 I CONTROL  [initandlisten] MongoDB starting : pid=3284 port=27017 dbpat
h=C:\data\db\ 64-bit host=DESKTOP-TFK928I
2017-09-13T04:02:57.863+1000 I CONTROL  [initandlisten] targetMinOS: Windows 7/Windows Server 2008 R
2
2017-09-13T04:02:57.864+1000 I CONTROL  [initandlisten] db version v3.4.7
2017-09-13T04:02:57.864+1000 I CONTROL  [initandlisten] git version: cf38c1b8a0a8dca4a11737581beafef
4fe120bcd
2017-09-13T04:02:57.865+1000 I CONTROL  [initandlisten] allocator: tcmalloc
2017-09-13T04:02:57.865+1000 I CONTROL  [initandlisten] modules: none
2017-09-13T04:02:57.866+1000 I CONTROL  [initandlisten] build environment:
2017-09-13T04:02:57.867+1000 I CONTROL  [initandlisten]     distmod: 2008plus
2017-09-13T04:02:57.867+1000 I CONTROL  [initandlisten]     distarch: x86_64
2017-09-13T04:02:57.868+1000 I CONTROL  [initandlisten]     target_arch: x86_64
2017-09-13T04:02:57.869+1000 I CONTROL  [initandlisten] options: {}
2017-09-13T04:02:57.876+1000 I -        [initandlisten] Detected data files in C:\data\db\ created b
y the 'wiredTiger' storage engine, so setting the active storage engine to 'wiredTiger'.
2017-09-13T04:02:57.877+1000 I STORAGE  [initandlisten] wiredtiger_open config: create,cache_size=76
56M,session_max=20000,eviction=(threads_min=4,threads_max=4),config_base=false,statistics=(fast),log
=(enabled=true,archive=true,path=journal,compressor=snappy),file_manager=(close_idle_time=100000),ch
eckpoint=(wait=60,log_size=2GB),statistics_log=(wait=0),
2017-09-13T04:02:58.127+1000 I CONTROL  [initandlisten]
2017-09-13T04:02:58.127+1000 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled fo
r the database.
2017-09-13T04:02:58.128+1000 I CONTROL  [initandlisten] **          Read and write access to data an
d configuration is unrestricted.
2017-09-13T04:02:58.129+1000 I CONTROL  [initandlisten]
2017-09-13T04:02:58.436+1000 I FTDC     [initandlisten] Initializing full-time diagnostic data captu
re with directory 'C:/data/db/diagnostic.data'
2017-09-13T04:02:58.438+1000 I NETWORK  [thread1] waiting for connections on port 27017
```

Screenshot (1) “Mongod.exe” command

```
c:\MongoDB\bin>mongo.exe
MongoDB shell version v3.4.7
connecting to: mongodb://127.0.0.1:27017
MongoDB server version: 3.4.7
Server has startup warnings:
2017-09-11T20:09:44.517+1000 I CONTROL  [initandlisten]
2017-09-11T20:09:44.517+1000 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled
 for the database.
2017-09-11T20:09:44.518+1000 I CONTROL  [initandlisten] **          Read and write access to data
 and configuration is unrestricted.
2017-09-11T20:09:44.518+1000 I CONTROL  [initandlisten]
>
```

Screenshot (2) “Mongo.exe” command

2. Database setup

After the testing of connection with MongoDB is finished, the next step is to import the smartmeter data into the database. Due to the “.xlsx” format of data file, the data file should be convert into CSV file.

There are two sheets in smartmeter data file. The first sheet is the description of building conditions, such as id, type, the number of people, air-conditioning type and the number of air-conditioning. The second sheet is records about the energy consumption for every building from January 2013 to December 2013. Every row presents the energy consumption in one day and the consumptions are record every half hour as an attribute in the table. In the end of a row, the sum of energy consumption is recorded.

At the beginning of data analysis, I want to analyze the whole situation of average energy consumption according to different building types and different air-conditioning types to find the suitable air-conditioning type for different buildings. Therefore, I create three collections at the beginning of analysis to store smartmeter data. The first collection is to store descriptive information about data and the second collection is to store buildings’ details. The third collection is to store the record of energy consumption.

x_NMI	Building	number of people	pool	type of AC	No. of AC
90141	double story	5	no	Evap	1
16254	unit	1	no	RCAC	1
90165	unit	2	no	Evap	1
90157	Double story house	3	No	Reverse cycle	
90148	Double story house	6	No	Evaporative, Reverse cycle	Three
90085	Double story house	2	No	None	None
90093	Double story house	4	No	Reverse cycle	Three

Screenshot (3) The second collection importing file

x_NMI	Estim	MeterRegActi	MtrRegAc	MtrRegAc	MtrRegAc	MtrRegAc	MtrRegAc	MtrRegAc	MtrRegAc	MtrRegAc	MtrRegAc	M
90000	0	12/01/2013	0.019	0.081	0.069	0.087	0.075	0.025	0.025	0.069	0.063	
90000	0	14/01/2013	0.037	0.037	0.031	0.069	0.094	0.037	0.031	0.037	0.037	
90000	0	2/01/2013	0.044	0.019	0.019	0.025	0.075	0.031	0.013	0.019	0.037	
90000	0	11/01/2013	0.075	0.025	0.019	0.044	0.081	0.019	0.019	0.037	0.081	
90000	0	17/01/2013	0.087	0.044	0.025	0.031	0.063	0.087	0.037	0.025	0.031	
90000	0	9/01/2013	0.025	0.081	0.031	0.025	0.019	0.069	0.05	0.019	0.019	
90000	0	6/01/2013	0.056	0.056	0.081	0.081	0.025	0.019	0.056	0.069	0.019	
90000	0	25/01/2013	0.087	0.094	0.094	0.063	0.037	0.05	0.094	0.069	0.037	
90000	0	1/01/2013	0.013	0.019	0.013	0.075	0.044	0.013	0.019	0.019	0.075	
90000	0	22/01/2013	0.063	0.087	0.056	0.031	0.031	0.094	0.056	0.031	0.031	
90000	0	18/01/2013	0.037	0.087	0.075	0.031	0.031	0.1	0.081	0.087	0.087	
90000	0	23/01/2013	0.1	0.081	0.044	0.031	0.037	0.087	0.094	0.037	0.037	

Screenshot (4) The third collection importing file

In the process of importing file, the command:

“c:\MongoDB\bin\mongoimport.exe -d fit5141 -c characteristic –headerline –type csv –file d:\characteristic.csv”

“c:\MongoDB\bin\mongoimport.exe -d fit5141 -c characteristic –headerline –type csv –file d:\smart.csv”

“c:\MongoDB\bin\mongoimport.exe -d fit5141 -c characteristic –headerline –type csv –file d:\descriptiveinformation.csv”

```
c:\MongoDB\Server\bin>mongoimport.exe -d fit5141 -c smartdata --headerline --type csv --file d:\smart.csv
2017-09-11T20:29:40.723+1000    connected to: localhost
2017-09-11T20:29:43.713+1000    [###############.........] fit5141.smartdata    13.2MB/20.8MB (63.6%)
2017-09-11T20:29:45.422+1000    [########################] fit5141.smartdata    20.8MB/20.8MB (100.0%)
2017-09-11T20:29:45.423+1000    imported 74704 documents

c:\MongoDB\Server\bin>mongoimport.exe -d fit5141 -c characteristic --headerline --type csv --file d:\characteristic.csv
2017-09-11T20:33:07.896+1000    connected to: localhost
2017-09-11T20:33:07.912+1000    imported 212 documents
```

Screenshot (5) Importing files

In the future, I will sort the energy consumption data into different collections in accordance with buildings type and analyze the consumption with time and season.

3. Connection and analysis of data in R

According to tutorial materials, the first step of connecting MongoDB with Rstudio is installation of mongolite, a package in Rstudio. The next step it to import mongolite library and create connections with different collections.

```
#install mongolite package
install.packages("mongolite")

#import mongolite and create connection with mongodb
library(mongolite)

dbconnection1 = mongo("smartdata", url="mongodb://localhost/fit5141",)
dbconnection2 = mongo("characteristic",url="mongodb://localhost/fit5141",)
```

Screenshot (6) Installation of mongolite and create connections

After retrieving all data in characteristic and smartdata collections and viewing all data, I extract id, daily consumption column and date from original data frame through coding.

```
#retrieve all smartmeter data and characteristic of house
smartdata= dbconnection1$find()
characteristic = dbconnection2$find()

#view all data
View(smartdata)
View(characteristic)

#select three columns(x_NMI, MeterRegActiveReadDt and MtrRegActNetEngyDailyKwh) in data"smartdata" to create a new data
#frame that contains id, date and daily consumption
dailydata<-data.frame(smartdata$x_NMI,smartdata$MeterRegActiveReadingDt,smartdata$MtrRegActNetEngyDailyKwh)
```

Screenshot (7) Extraction of daily consumption data

smartdata.x_NMI	smartdata.MeterRegActiveReadingDt	smartdata.MtrRegActNetEngyDailyKwh
90000	14/01/2013	5.149
90000	2/01/2013	3.878
90000	11/01/2013	2.257
90000	9/01/2013	1.941
90000	6/01/2013	2.269
90000	12/01/2013	4.947
90000	1/01/2013	1.800
90000	17/01/2013	5.374
90000	25/01/2013	8.587
90000	13/01/2013	5.823
90000	27/01/2013	4.299
90000	24/01/2013	5.032
90000	5/01/2013	3.051
90000	21/01/2013	6.705
90000	7/01/2013	2.376
90000	29/01/2013	2.336
90000	20/01/2013	6.406

Screenshot (8) Daily consumption data

Then I installed new package “zoo” to aggregate the sum of energy consumption according to season and change the column name for readable convenience.

```
#install library "zoo" to get the sum of daily consumption of energy according to season for every building
install.packages("zoo")
library(zoo)

#calculate the sum of daily consumption of energy according to season for every building
seasondata<-aggregate(dailydata$smartdata.MtrRegActNetEngyDailyKwh~dailydata$smartdata.x_NMI + yearqtr, transform(daily

#change the colums name in seasondata
colnames(seasondata)[which(names(seasondata)=="dailydata$smartdata.MtrRegActNetEngyDailyKwh")]<- "seasonconsumption"
colnames(seasondata)[which(names(seasondata)=="dailydata$smartdata.x_NMI")]<- "x_NMI"
```

Screenshot (9) Season consumption data

Due to the various types of numbers and words in the same column, I format the content type through coding. There are some buildings in characteristic data frame, which is not in seasondata data frame. So the buildings are deleted to avoid coding and process errors.

```
#delete rows that contain none data in number of people column ,'type of AC' column and 'no.ofAC' column
char<-subset(characteristic, characteristic$`number of people` !="")
char<-subset(char, char$`type of AC` !="" & char$`type of AC` != "I don't know")
char<-subset(char, char$x_NMI != "90132" & char$x_NMI != "90019")

#convert string into number in 'no.ofAc' column
char$No[char$No == "One"] <- 1
char$No[char$No == "Two"] <- 2
char$No[char$No == "Three"] <- 3
char$No[char$No == "Four"] <- 4
char$No[char$No == "Five"] <- 5
char$No[char$No == "None"] <- 0
char<-subset(char, char$No != "")
char$`number of people`[char$`number of people` == "+1"] <-1 # correct a input mistake about data

#covert char to int in no of people and no of ac columns
char$`number of people`<-as.numeric(char$`number of people`)
char$No<- as.numeric(char$No$` of AC`)

#formate the different words with same meanings in 'type of AC' and 'building' columns
unique(char$`type of AC`)
char$`type of AC`[char$`type of AC` == "evap" | char$`type of AC` == "Evaporative"] <- "Evap"
char$`type of AC`[char$`type of AC` == "evap & rcac"|char$`type of AC` == "evap & RCAC" | char$`type of AC` == "Evapora
char$`type of AC`[char$`type of AC` == "Reverse cycle"] <-"RCAC"
unique(char$Building)
char$Building[char$Building == "unit"] <- "Unit"
char$Building[char$Building == "singel level" | char$Building == "single leve" | char$Building == "single level"] <- "S
char$Building[char$Building == "double story"]<- "Double story house"
char$Building[char$Building == "apartment"]<- "Apartment"

#firstly delete pool column due to it does not affect the consumption of energy a lot
charnopool<-char
charnopool$pool<-NULL

#delete negative consumption value in seasondata and the consumption without house detail
#delete buildings without consumption information in chairnopool
seasondata <-subset(seasondata, seasondata$seasonconsumption >= 0)
seasondata<-subset(seasondata,seasondata$x_NMI %in% unique(charnopool$x_NMI))
charnopool<-subset(charnopool,charnopool$x_NMI %in% unique(seasondata$x_NMI))
```

Screenshot (10) formatting data

After formatting data, I calculated the average energy consumption of per person with one air-conditioning and plot data according to air-conditioning types or the combine of air-conditioning types and building types.

```r
#calculate the average consumption of per person in one AC
seasondataaverage<-seasondata
np=0
air=0
ids<-unique(charnopool$x_NMI)

for (i in ids){
  air <- charnopool$No[charnopool$x_NMI == i]
  np <- charnopool$`number of people`[charnopool$x_NMI == i]
  if (air > 0){
    seasondataaverage$seasonconsumption[seasondataaverage$x_NMI == i]<-seasondataaverage$seasonconsumption[seasondataaverage$x_NMI == i]/np/air
  }
  else{
    seasondataaverage$seasonconsumption[seasondataaverage$x_NMI == i]<-seasondataaverage$seasonconsumption[seasondataaverage$x_NMI == i]/np
  }
}
```

Screenshot (11) Average energy consumption

```r
install.packages("ggplot2")
library(ggplot2)

#plot the consumption energy in each season for per person
ggplot(seasondataaverage, aes(x=ACtype,y=seasondataaverage$seasonconsumption)) + geom_point(stat="identity", color = "red")+
  ggtitle("The average energy consumption in each season for per person")+
  labs(x="Air Conditioning types", y= "Energy Consumption")

#plot the consumption energy sorted by season for per person
seasondisplay<- seasondataaverage
seasondisplay$Season <- ""
seasondisplay$Season[seasondisplay$yearqtr == "2013 Q4"] <-"Spring"
seasondisplay$Season[seasondisplay$yearqtr == "2013 Q1"] <-"Summer"
seasondisplay$Season[seasondisplay$yearqtr == "2013 Q2"] <-"Autumn"
seasondisplay$Season[seasondisplay$yearqtr == "2013 Q3"] <-"Winter"
ggplot(seasondisplay, aes(x=ACtype,y=seasonconsumption, fill = Season)) + geom_bar(stat="identity", color = "white")+
  ggtitle("The average energy consumption sorted by season for per person")+
  labs(x="Air Conditioning types", y= "Energy Consumption")+
  theme_minimal()

#plot consumption according to building type and ACtype
house<-seasondataaverage
house$type <-""
for (i in unique(charnopool$x_NMI)){
  house$type[house$x_NMI == i] <- charnopool$Building[charnopool$x_NMI==i]
}
house$combine <- paste(house$type,house$ACtype)
ggplot(house, aes(x=combine,y=seasonconsumption)) + geom_point(stat="identity", color = "red")+
  ggtitle("The average energy consumption for per person")+
  labs(x="Building and Air Conditioning types", y= "Energy Consumption")+
  theme(axis.text.x = element_text(angle =  90, vjust = 0.5))
```

Screenshot (12) ggplot 2 to plot energy consumption

In figure 1, the average energy consumption of RCAC is the highest which is reasonable. However, it seems that the average consumption of buildings without air-conditioning is higher than the buildings with Evap&RACA and the consumption of Evap is higher than Evap&RACA, which are anomalies. Therefore, we should do more analysis about these anomalies in the future.

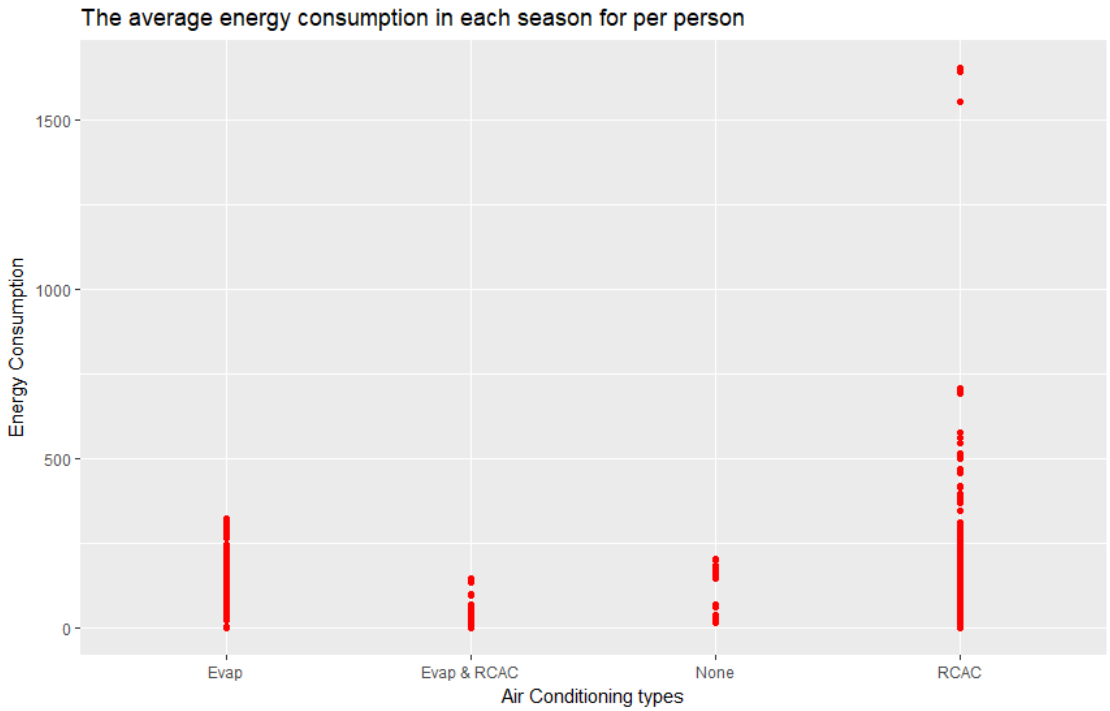


Figure 1

The figure 2 shows the average energy consumption of different buildings with various air-conditioning. The energy consumption of single level house with RACA is the highest and the next is the unit with RACA. On the other hand, the energy consumption of double story house with Evap is higher than that with RACA and Evap. Therefore, it assumes that RACA&Evap is the most suitable AC types for double story house. Finally, other buildings with different AC

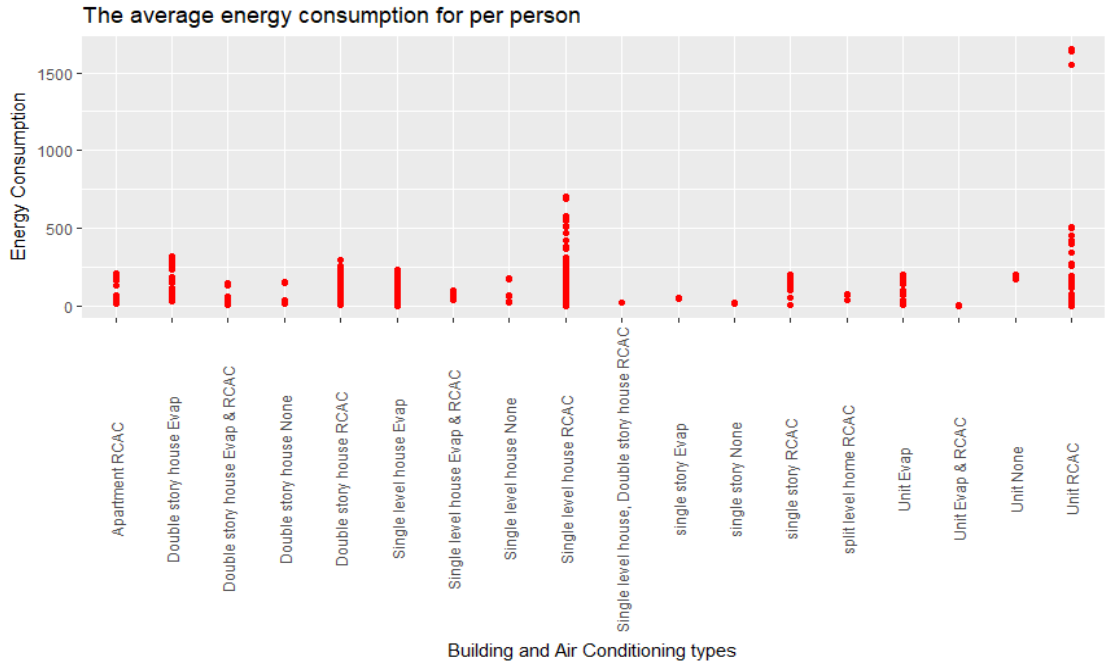


Figure 2

type subscribe the predicted energy consumption levels, such as double story house consumes more energy than single story house.

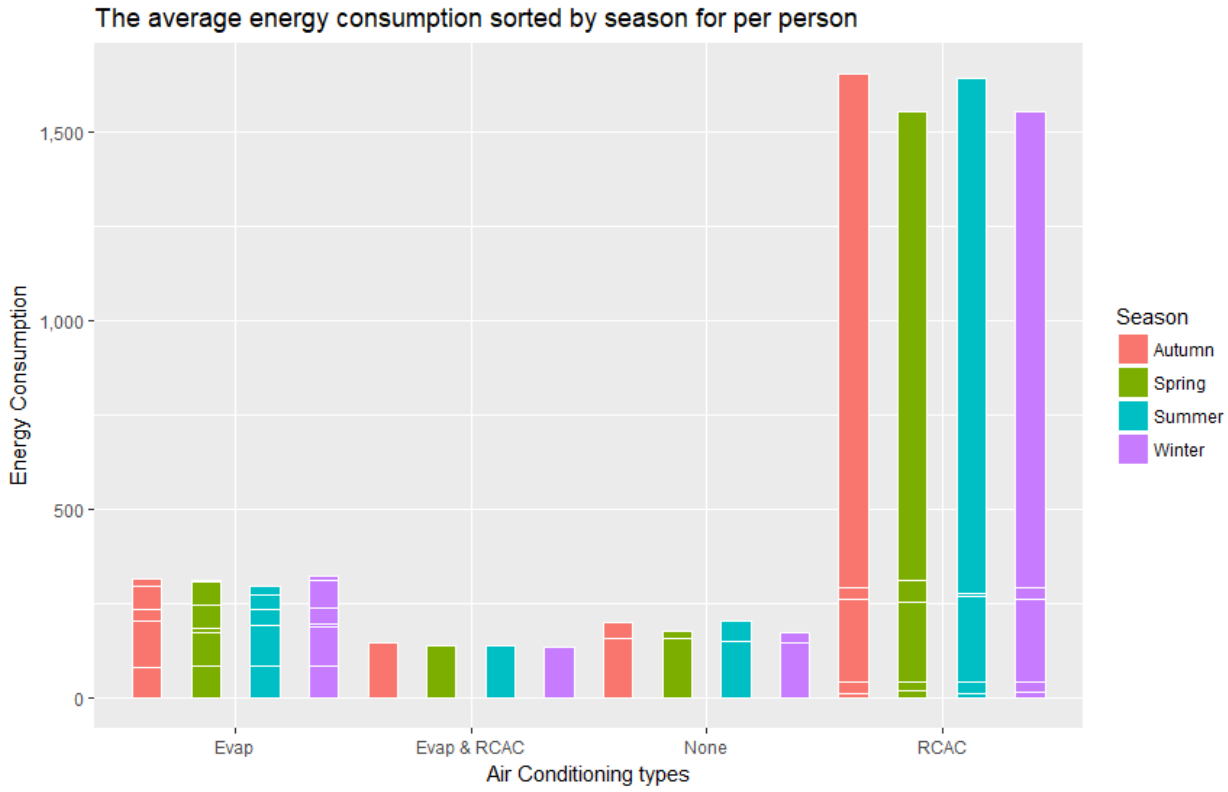


Figure 3

According to figure 3, the energy consumption of buildings with different AC types in Autumn is higher than other seasons and RCAC buildings consumed the extremely most energy in 4 seasons. Because there are huge gaps about the number of buildings, the analysis will focus on the energy consumption in different time in one day and daily consumption in one season.

4.The import of wrangled data

After wrangling data, several new collections were created to store wrangled data by command “db.createCollection (“name”, (capped: true, size: xxx, max : xxxx). In the import process, there are some problems about data type in column “yearqrt”, so the yearqrt column was convert to season column whose type is character to express 4 different seasons.

```
c:\MongoDB\Server\bin>mongo.exe
MongoDB shell version v3.4.7
connecting to: mongodb://127.0.0.1:27017
MongoDB server version: 3.4.7
Server has startup warnings:
2017-09-11T20:22:34.057+1000 I CONTROL  [initandlisten]
2017-09-11T20:22:34.057+1000 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2017-09-11T20:22:34.058+1000 I CONTROL  [initandlisten] **          Read and write access to data and configuration is u
nrestricted.
2017-09-11T20:22:34.060+1000 I CONTROL  [initandlisten]
> db.c
db.changeUserPassword(  db.commandHelp(         db.createCollection(    db.createView(
db.cloneCollection(     db.constructor          db.createRole(          db.currentOP(
db.cloneDatabase(       db.copyDatabase(        db.createUser(          db.currentOp(
> db.createCollection("char",{ capped: true, size:50000000, max : 5000})
{ "ok" : 1 }
> db.createCollection("charnopool",{ capped: true, size:50000000, max : 5000})
{ "ok" : 1 }
> db.createCollection("season",{ capped: true, size:50000000, max : 5000})
{ "ok" : 1 }
> db.createCollection("average_season",{ capped: true, size:50000000, max : 5000})
{ "ok" : 1 }
> db.createCollection("ave_season_building",{ capped: true, size:50000000, max : 5000})
```

```
#import wrangled data to Mongodb
seasondata<- seasondataaverage
seasondata$Season <- ""
seasondata$Season[seasondata$yearqtr == "2013 Q4"] <-"Spring"
seasondata$Season[seasondata$yearqtr == "2013 Q1"] <-"Summer"
seasondata$Season[seasondata$yearqtr == "2013 Q2"] <-"Autumn"
seasondata$Season[seasondata$yearqtr == "2013 Q3"] <-"Winter"
seasondata$yearqtr<-NULL
house$Season <- ""
house$Season[house$yearqtr == "2013 Q4"] <-"Spring"
house$Season[house$yearqtr == "2013 Q1"] <-"Summer"
house$Season[house$yearqtr == "2013 Q2"] <-"Autumn"
house$Season[house$yearqtr == "2013 Q3"] <-"Winter"
house$yearqtr<-NULL

dbconnection3 = mongo("char", url="mongodb://localhost/fit5141",)
dbconnection4 = mongo("charnopool",url="mongodb://localhost/fit5141",)
dbconnection5 = mongo("season", url="mongodb://localhost/fit5141",)
dbconnection6 = mongo("average_season",url="mongodb://localhost/fit5141",)
dbconnection7 = mongo("ave_season_building", url="mongodb://localhost/fit5141",)
dbconnection3$insert(char)
dbconnection4$insert(charnopool)
dbconnection5$insert(seasondata)
dbconnection6$insert(seasondisplay)
dbconnection7$insert(house)
```

Screenshot (13) import wrangled data

5.Tableau connection

Among various ODBCs, I choose Simba ODBC to connect with MongoDB in Tableau. After installing Simba ODBC and configuration of it, open tableau and choose other ODBC to connect with MongoDB.

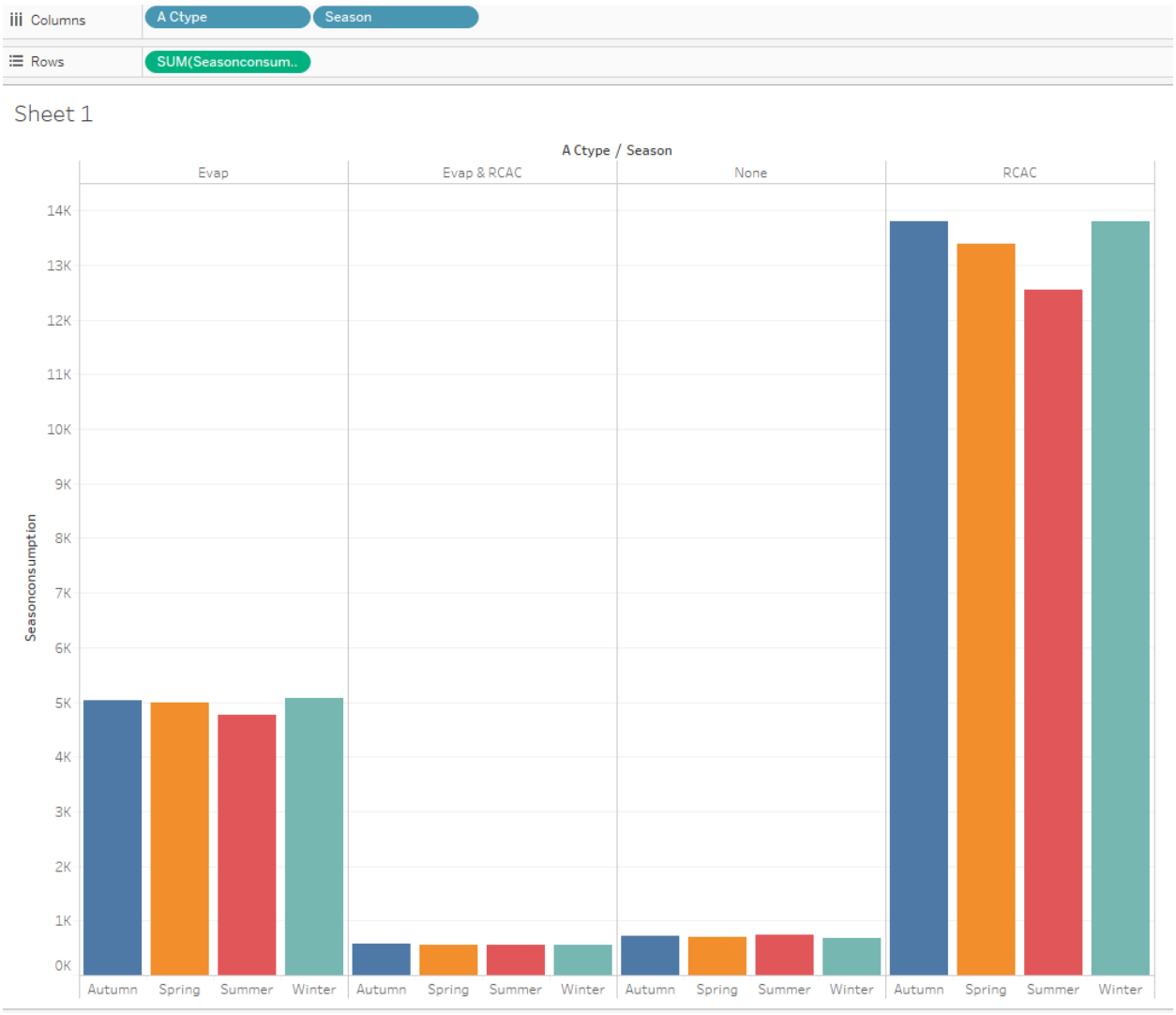
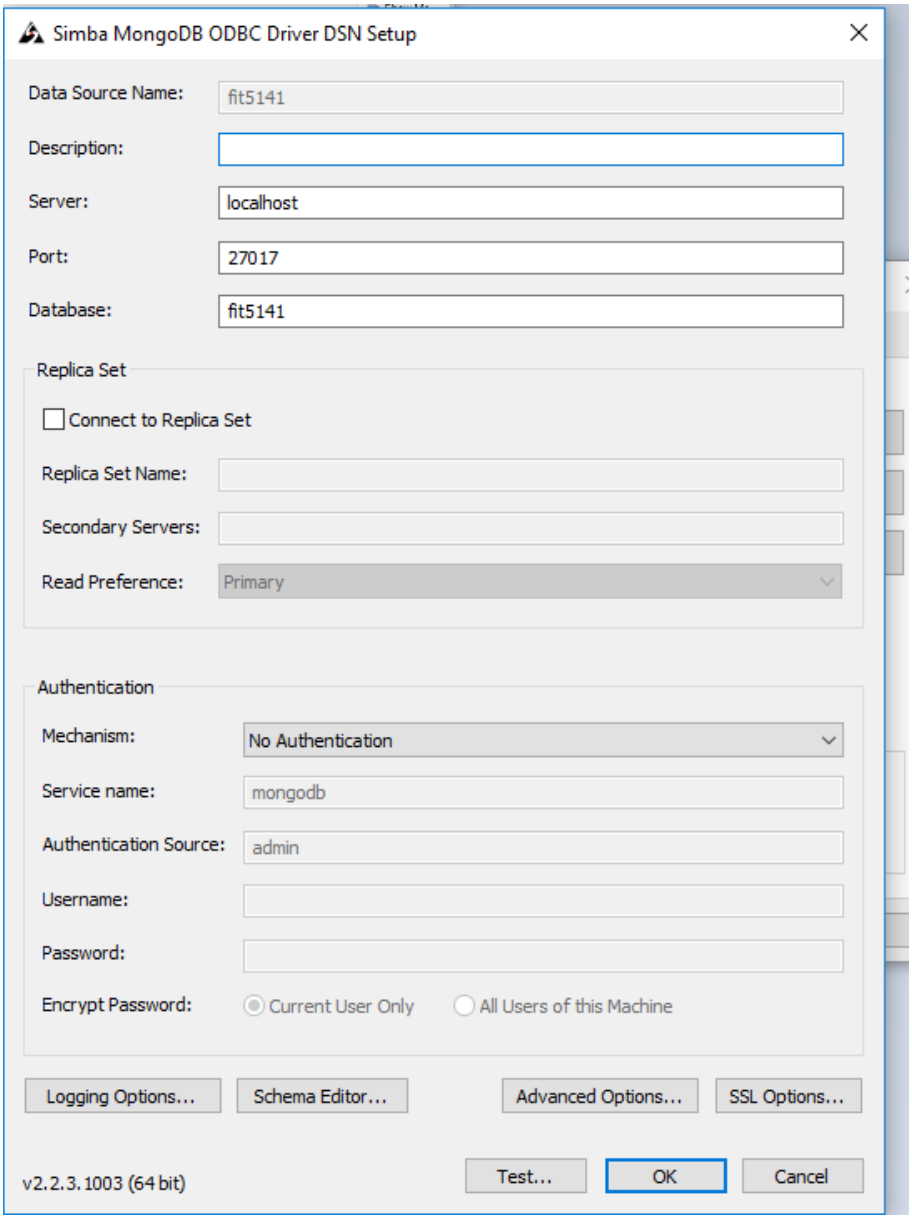


Figure 4 Season analysis in Tableau



Screenshot (14) ODBC configuration