설치

본 서버에서 사용된 환경은 밑의 표와 같다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | OS | Language | Package | 기타 |
| Server | Ubuntu 12.04 | Node.js | coap 0.21.0, mongo-to-csv 1.0.1, mongodb 3.0.4,  node-schedule 1.3.0 | MongoDB 3.0.4 |
| Data Analysis | Windows 10 | R | Lubridate 1.7.1, ggplot2 2.2.1, corrplot 0.8.4 | RStudio |

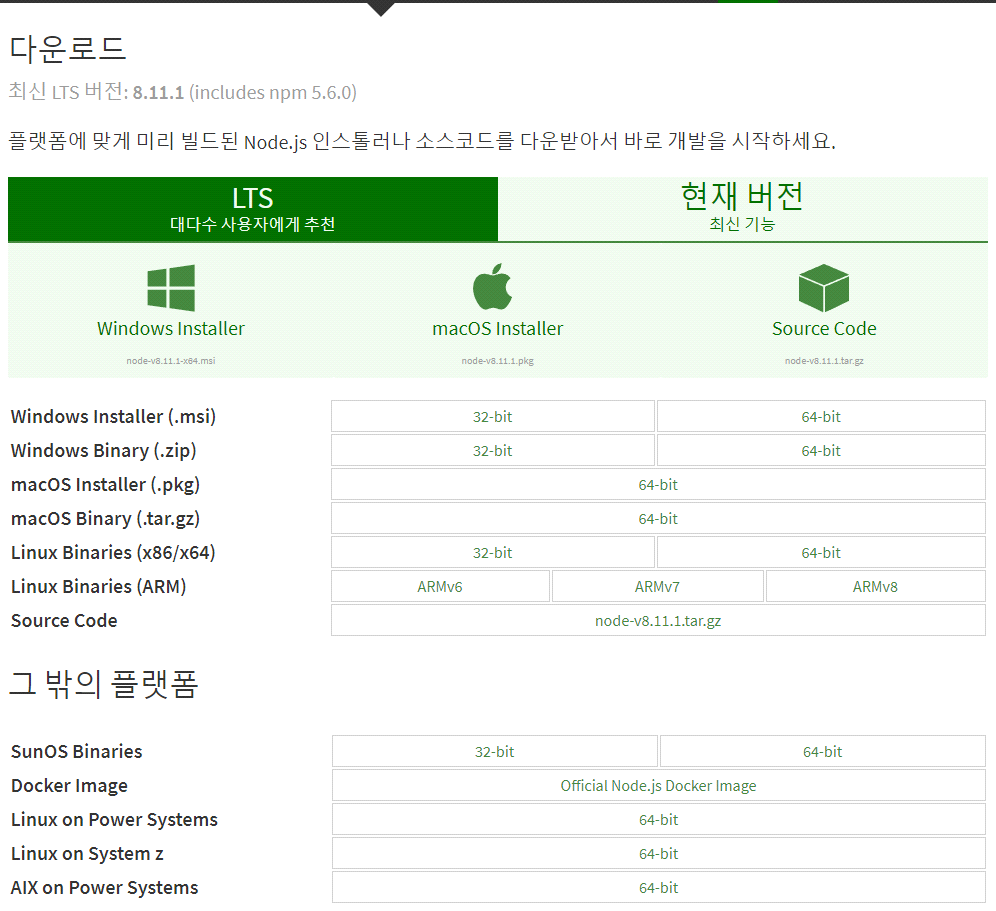
<표 x. 실험 환경>

**3.2 IoT 빅데이터 수집 및 처리를 위한 서버**

본 실험에서 빅데이터 수집 및 처리를 위한 서버는 Ubuntu 12.04에서 실행된다. <표 x>에서 보듯 서버를 실행하기 위해서는 Node.js 플랫폼 , coap , mongodb , mongo-to-csv , node-schedule 모듈, MongoDB Server 프로그램이 필요하다.

**1.Node.js**

1. <https://nodejs.org/ko/download/> 에 들어가면 밑의 그림과 같이 다운로드 페이지가 뜬다.



2. 자신의 운영체제에 맞는 파일을 다운받는다. 본 서버에서는 Ubuntu 12.04를 사용하고 있어 Linux Binaries(x64)를 다운받았다.

3. $tar -xvf node-v8.10.0-linux-x64.tar.xz 명령으로 압축파일을 풀고 디렉토리에 들어가면 bin 디렉토리에 node있는 node나 npm로 node.js를 실행시키거나 필요한 패키지를 다운로드 할 수 있다.

4. 다음의 명령어로 nodejs 디렉토리를 /usr 디렉토리로 복사한다.

 $mv node-v8.10.0-linux-x64 node

 $sudo cp -r node/{bin,include,lib,share} /usr/

5.npm 명령어로 본 서버에서 사용된 패키지들을 다운로드한다.

$npm install coap --save

$npm install mongo-to-csv --save

$npm install mongodb --save

$npm install node-schedule --save

**MongoDB**

MongoDB의 경우 공식 홈페이지에 있는 웹페이지를 기반으로 설치했다. 사이트는 다음과 같다.

<https://docs.mongodb.com/v3.0/tutorial/install-mongodb-on-ubuntu/>

1.package management system에 이용되는 공인키를 import한다. 우분투 패키지 관리 툴 dpkg나 apt는 GPG keys를 사용해 패키지에 서명하도록 요구한다.

$sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv 7F0CEB10

2.MongoDB 리스트를 추가한다.

/etc/apt/sources.list.d/mongodb-org-3.0.list 리스트 파일을 만들어서 apt 저장소의 업데이트가 가능하게 한다.

$echo "deb [http://repo.mongodb.org/apt/ubuntu precise/mongodb-org/3.0](http://repo.mongodb.org/apt/ubuntu%20precise/mongodb-org/3.0) multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-3.0.list

문서에도 나와있지만 본 서버는 우분투 12.04를 사용하므로 precise다. 만약에 버전이 다르면 그 버전에 맞는 명령어가 필요하다.

3.apt 서버를 업데이트한다.

$sudo apt-get update

4.MongoDB 패키지를 다운로드한다.

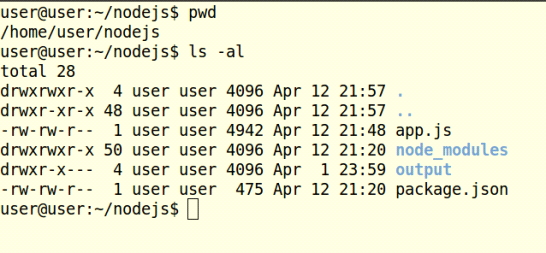
$sudo apt-get install -y mongodb-org 명령어로 일반적으로 다운로드 할 수도 있고

sudo apt-get install -y mongodb-org=3.0.15 mongodb-org-server=3.0.15 mongodb-org-shell=3.0.15 mongodb-org-mongos=3.0.15 mongodb-org-tools=3.0.15 명령어로 특정 버전을 사용할 수도 있다. 본 서버에서는 nodejs의 mongodb 패키지의 버전과 MongoDB 버전을 맞추기 위해 밑의 명령어를 사용해서 버전을 맞췄다.

5.MongoDB 서버를 실행시킨다.

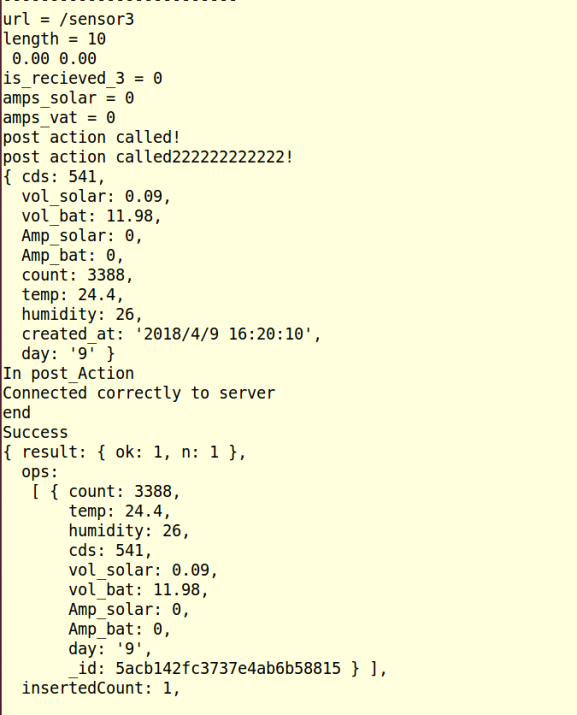
$sudo service mongod start

앞의 과정을 수행해 Node.js , npm package들, MongoDB Server를 설치하게 되면 다음과 같이 app.js , node\_module , package.json 파일이 디렉토리에 있게 된다. 아니면 Node.js만 설치한 뒤, CD에 첨부된 Server.tar.gz 파일을 tar –zxvf Server.tar.gz 명령어로 풀어도 된다.



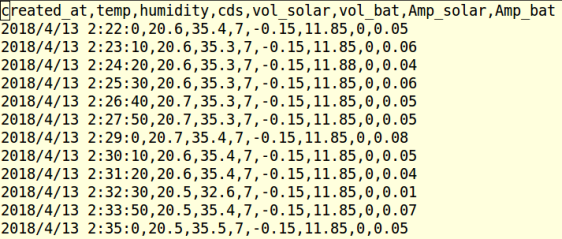
<그림 x. 서버의 디렉토리>

그 상태에서 $node app.js 명령을 실행하게 되면



<그림 x. 서버 실행 결과>

다음과 같은 결과가 나온다. 그 상태에서 자정이 지나면 app.js가 있던 폴더에서 output/월/일.csv 파일이 생성된다. 이 csv 파일은



<그림 x. 자정이 되면 만들어지는 csv 파일>

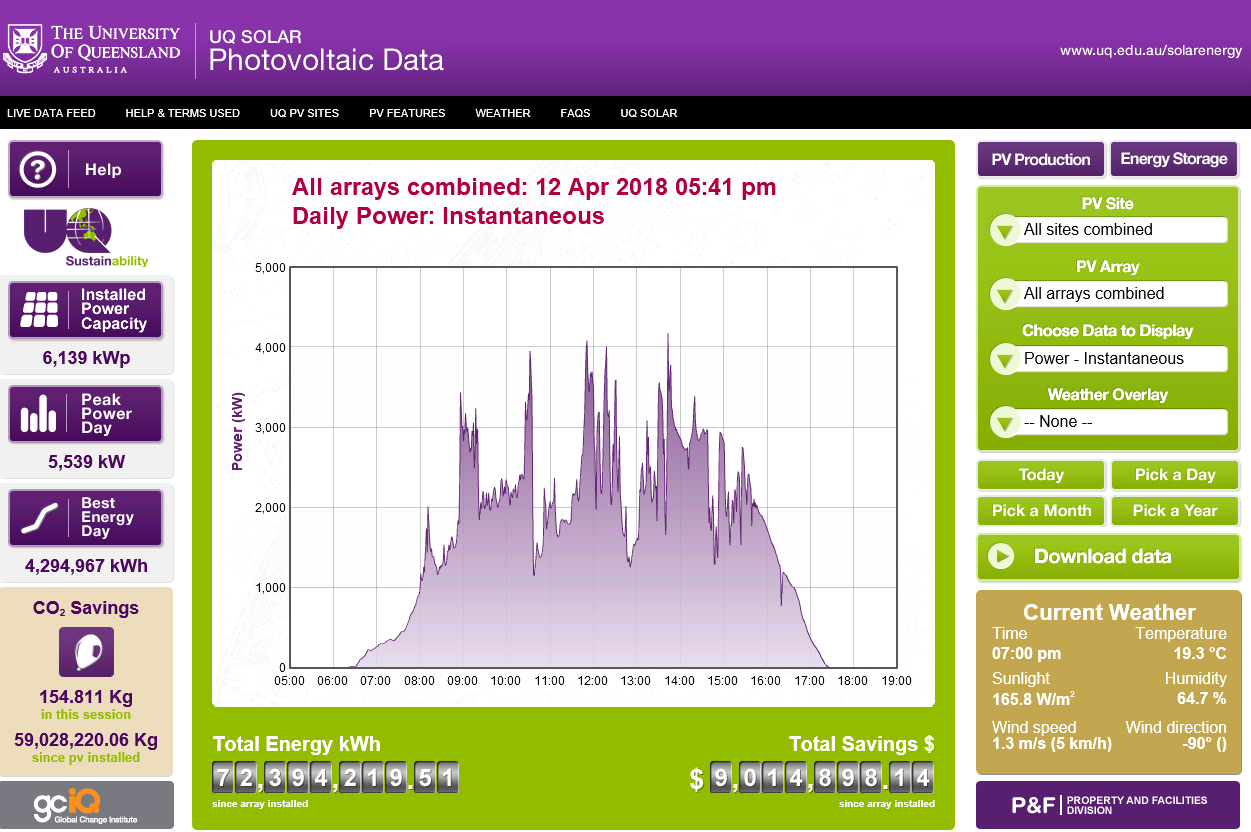
**4. 데이터 분석 시스템**

본 연구에서는 데이터 분석을

1. 호주 The University Of Queensland에서 제공한 태양열 에너지 데이터를 참고하여 시간별 평균 에너지 생산량 출력
2. 호주 The University Of Queensland에서 제공한 태양열 에너지 데이터를 참고하여 각 환경변수 간의 상관계수분석
3. 호주 The University Of Queensland에서 제공한 태양열 에너지 데이터를 참고하여 월별 평균 에너지 생산량 출력
4. IoT 센싱 디바이스로 수집한 데이터를 참고하여 시간별 평균 에너지 생산량 출력
5. IoT 센싱 디바이스로 수집한 데이터를 참고하여 각 환경변수 간의 상관계수분석

이렇게 4번을 실시한다. 데이터 분석을 위한 툴을 다운로드 하기 전에, 1) 와 2)에서 사용되는 호주 The University Of Queensland에서 제공한 데이터를 다운로드한다.

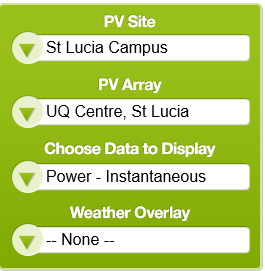
호주의 데이터는 <http://solar.uq.edu.au/user/reportPower.php> 에 가면 얻을 수 있다.



<그림 x. University Of Queensland 사이트>

University Of Queensland에서는 하나의 파일에 전력에너지, 기후정보가 다 들어가 있는 것이 아닌,전력정보(Power , Energy , hour)가 들어있는 파일 , 기후정보(Airtemp , “airtemp” , “humidity” , “insolation” ,“windspeed” , “winddirection”)가 들어있는 파일, 총 2개가 있다.

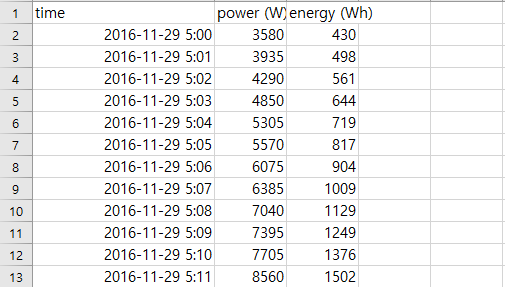
단, 현재는 St Lucia Campus밖에 기후정보를 얻지 못한다. 따라서 St Lucia Campus의 UQ Centre, St Lucia 정보만 얻어서 분석을 수행한다.



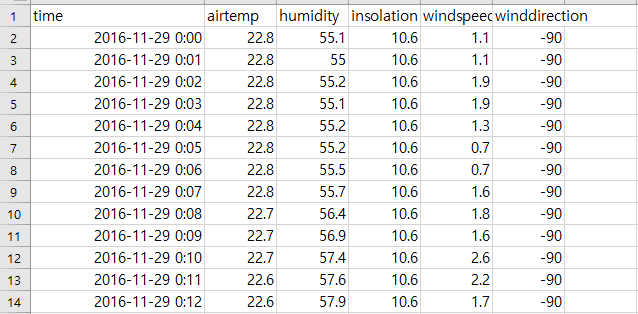
<그림 x. 지역정보 선택>

University Of Queensland 사이트에서 위의 우측에 있는 지역을 선택하는 란에서 <그림x>와 같이 지역을 선택하고 우측 하단에 Download data 버튼을 누르고 Download Daily Logs 를 누른 뒤, 날짜를 선택하면 그 날짜에 해당하는 정보를 csv 파일로 저장할 수 있다. 전력량 데이터와 기후 데이터가 합쳐진게 아닌 따로 다운로드 해야한다. 전력량은 pe , 기후 데이터는 w로 시작한다.

밑의 <그림 x>는 각각 pe20161129-20171129.csv로 전력량을 나타내는 csv 파일, w20161129-20171129.csv로 기후 정보를 나타내는 csv 파일이다.



<그림 x. pe20161129-20171129.csv>



<그림 x. w20161129-20171129.csv>

**R**

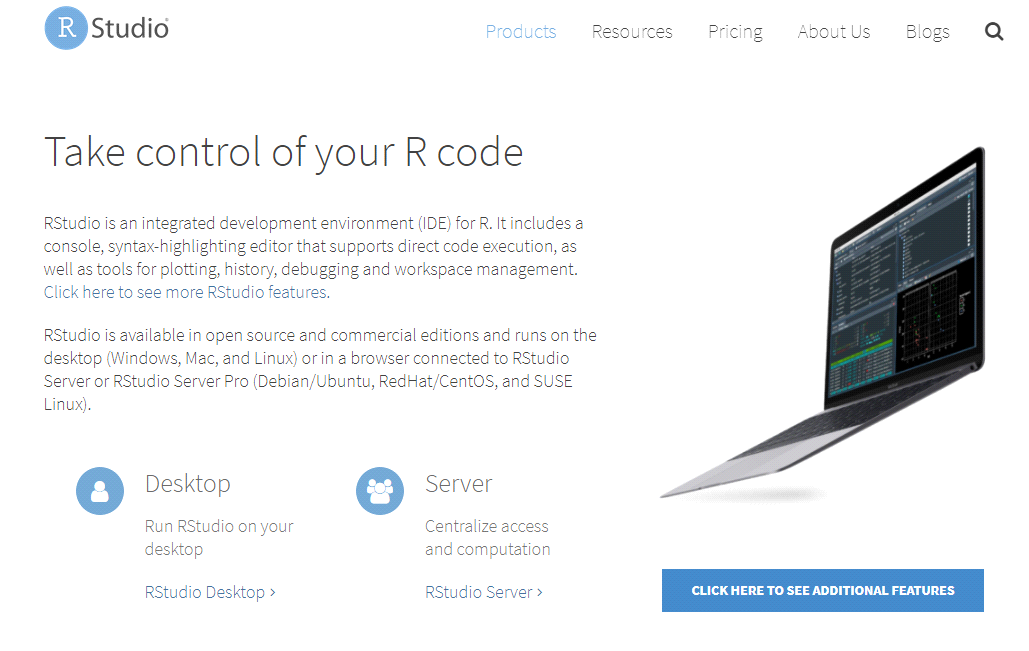
호주 데이터를 얻은 뒤, 분석용 툴을 사용할 수 있는 R을 설치한다. 리눅스의 R은 최신버전을 깔기 힘들고 패키지를 다운로드 하는데 에로사항이 있어 통계분석은 윈도우로 진행하였다. 따라서 본 문서에서는 윈도우 환경에서 R을 설치한다.

R 같은 경우에는 프로그래밍 언어인 R과 컴파일 환경인 RStudio를 깔아야한다. R은 공식사이트 <https://www.r-project.org/>에서 다운로드 할 수 있다.

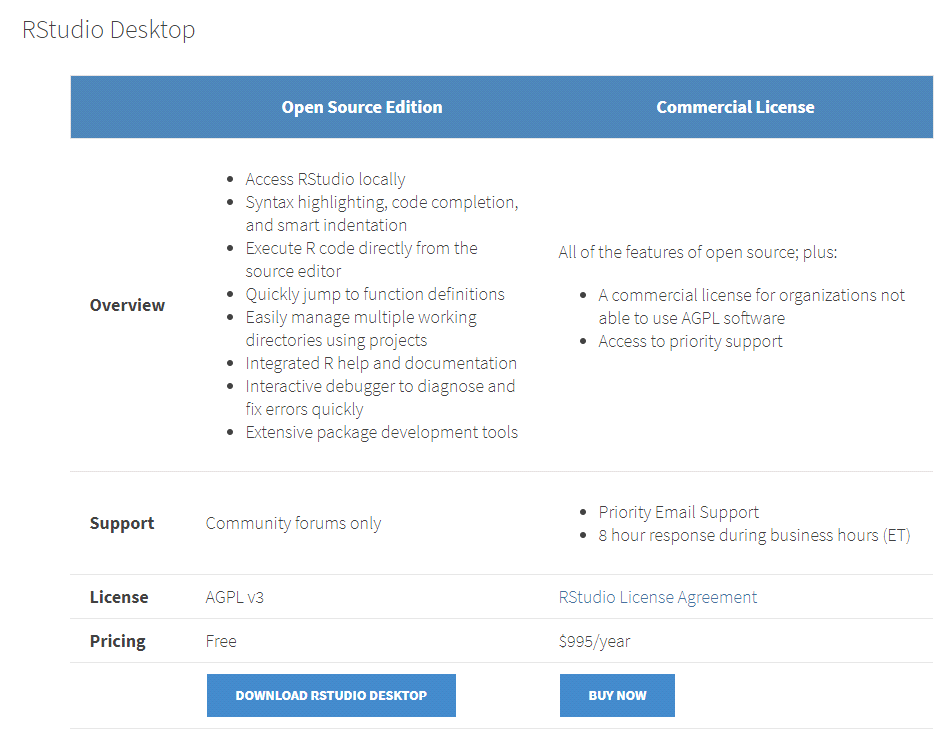


그 다음은 RStudio를 설치한다.

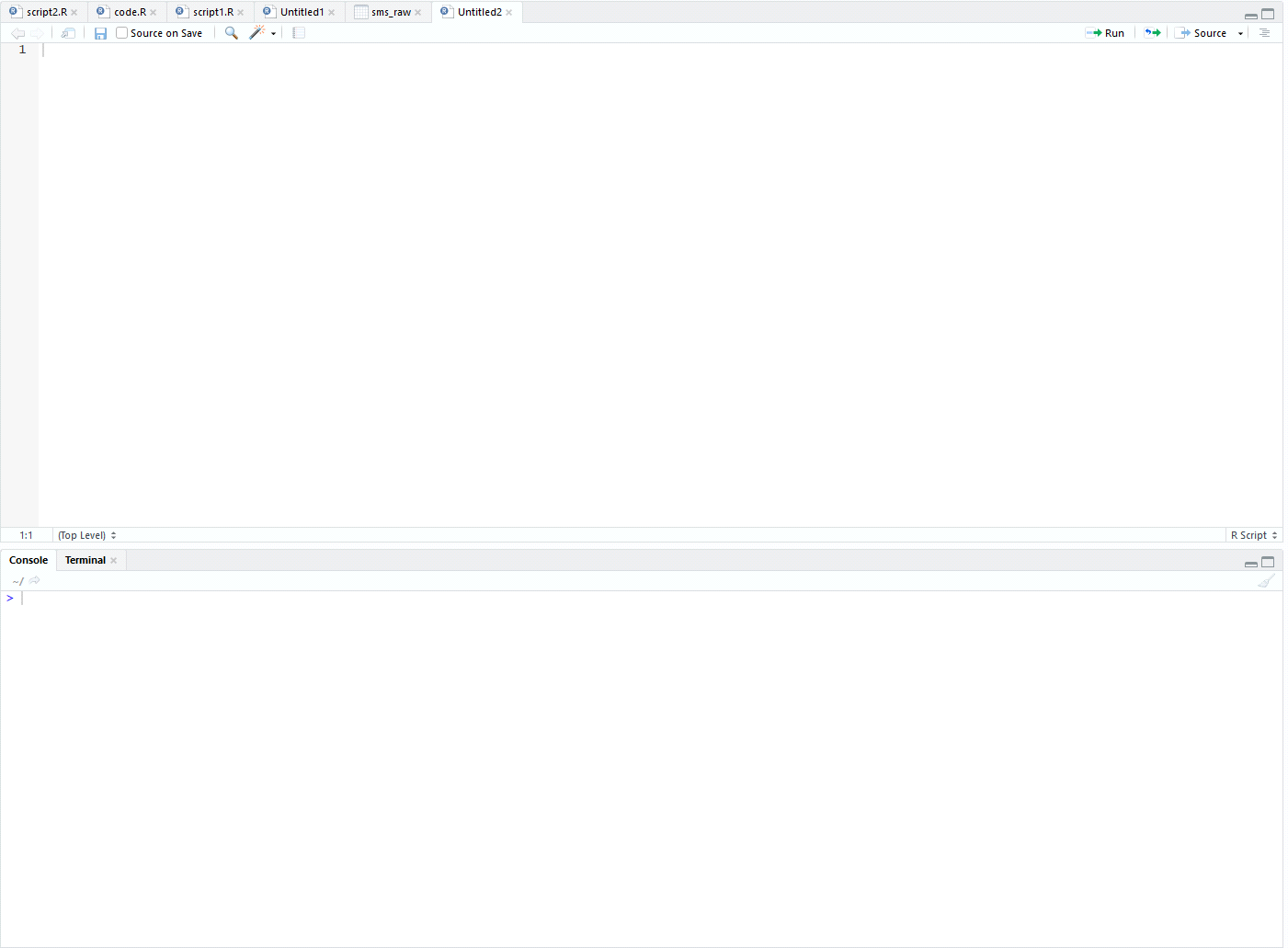
<https://www.rstudio.com/> 홈페이지에서 Products -> RStudio를 들어가면 다음과 같은 웹페이지가 뜬다.



Desktop을 선택하면 다운로드 창으로 이동하는데 거기서 자신이 다운로드 받고 싶은 버전을 다운로드 하면 된다.



RStudio에 기본적으로 깔려있는 패키지로는 상관분석, 시간데이터 관리, 그래프 출력 등은 불가능하다. 따라서 RStudio 내에서 추가적으로 설치를 해야한다. RStudio를 키면 기본적으로 나오는 화면에서 좌측 하단의 콘솔 창에서



install.packages("ggplot2")

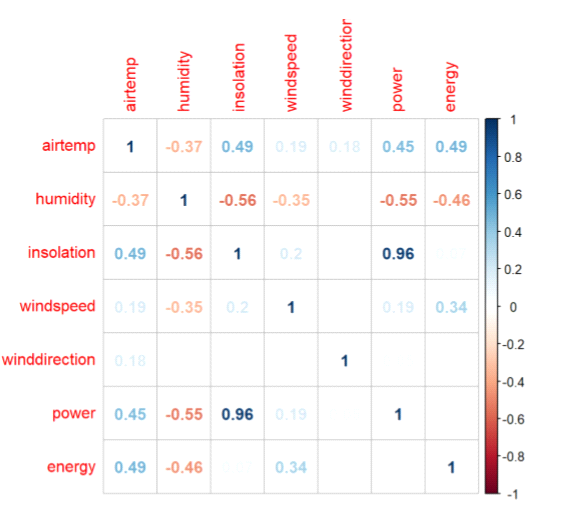
install.packages("corplot")

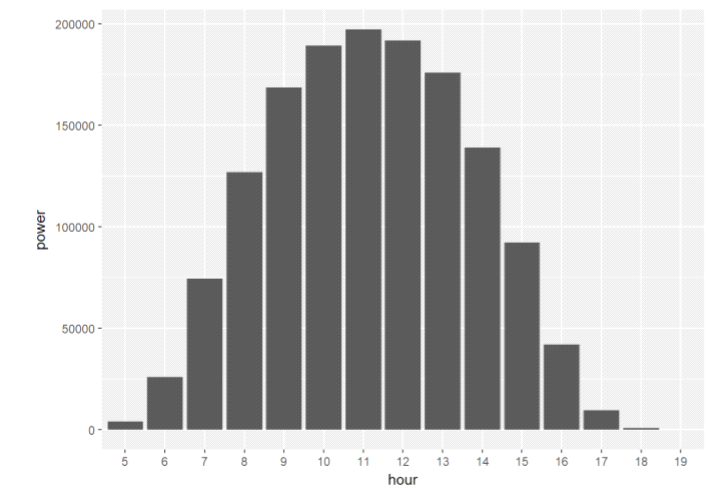
install.packages("lubridate")

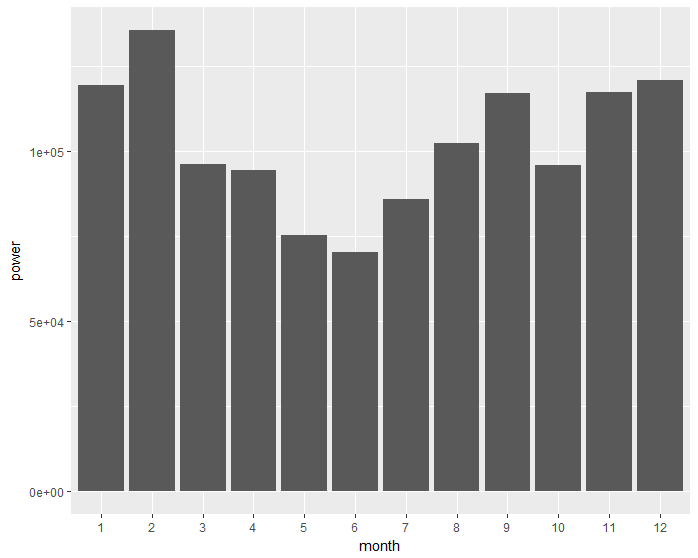
를 쳐서 패키지를 다운로드한다.

------------------------------------------------------------------------

호주 The University Of Queensland에서 제공한 태양열 에너지 데이터를 참고하여 시간별 평균 에너지 생산량 출력을 하고 싶다면 CD에 첨부되어 있는 Program -> Data Analysis -> Australia\_energy.R 파일에 있는 소스코드를, The University Of Queensland에서 제공한 태양열 에너지 데이터를 참고하여 환경변수간의 상관계수를 구하고 싶으면 같은 폴더에 있는 Australia\_cor.R 파일을 실행시키면 된다. 이때 R 코드에 앞에서 구한 The University Of Queensland의 데이터가 있는 폴더를 정확하게 fread 함수에서 써야 한다.

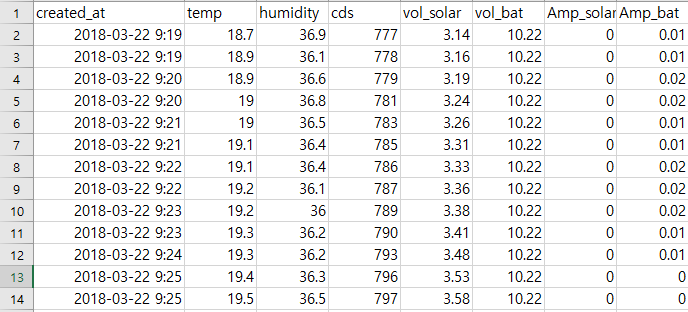






첫번째 그림은 Australia\_cor.R을 실행시켰을 때의 결과이고 두번째 그림은 Australia\_energy.R 파일을 실행시켰을 때 결과 , 세번째는 Australia\_month\_energy.R을 실행시켰을 때의 결과이다.

다음으로 3) , 4)인 IoT 센싱 디바이스로 데이터를 수집해 이전과 마찬가지로 시간별 평균 전력생산량 , 각 환경변수 간의 상관관계분석을 실시한다.



<그림 x.IoT 센싱 디바이스로부터 받은 전압값과 환경변수>

<그림 x>에서 보다시피 아두이노로부터 받은 csv 파일에는 다음과 같이 8개의 필드로 이루어져 있는데 이 csv 파일을 R로 읽어들여 데이터 분석을 한다.

아두이노로부터 받은 데이터를 가지고 시간당 평균 전압생성량을 그래프로 표시할거면 CD에 첨부되어 있는 Program -> Data Analysis -> IoTServer\_Energy.R 파일에 있는 소스코드를, 환경변수간의 상관계수를 구하고 싶으면 같은 폴더에 있는 IoTServer\_cor.R 파일을 실행시키면 된다. 이때도 마찬가지로 R 코드에 앞에서 구한 IoT 디바이스의 환경변수가 저장되어 있는 csv 파일이 있는 폴더를 정확하게 fread 함수에서 써야 한다.

