R-ohjelmiston lisäpaketit ja data

Vesa Pekkanen

Matti Kiiski

14. syyskuuta 2012

Tämä on lyhyt johdatus R-ohjelmiston kirjastojen käyttöönottoon ja datan lukemiseen. Hyödyllisiä komentoja:

- **help.start**() avaa ohjesivuston
- ?**funktion.nimi**() avaa funktiomääritelmän.

Jos ei ole aikaisempaa kokemusta R-ohjelmistosta, kannattaa katsoa ohjeen lopussa olevat esimerkit graafisen käyttöliittymän ja komentorivin käytöstä. Opas on kirjoitettu R-ohjelmiston virallisen manuaalin pohjalta: `http://cran.r-project.org/doc/manuals/R-admin.html`

Sisältö

Kirjastot

Kirjastoista puhuttaessa on hyvä käyttää oikeaa terminologiaa. *Paketti* ladataan *kirjastosta* funktiolla **library**(). Kirjasto on siis hakemisto, joka sisältää asennettuja paketteja. R-ohjelmiston asennuksen yhteydessä luotava oletuskirjasto on **R_HOME/library**. Voit tarkastaa tämän komennolla .**libPaths**().

Kirjastojen hallinta

R-paketit ovat asennettuina *kirjastoihin*, jotka ovat tiedostojärjestelmän hakemistoja. Ne sisältävät alihakemistoja, joissa asennetut paketit sijaitsevat.

R-ohjelmiston mukana tuleva kirjasto **R_HOME/library** on olion .**Library** arvo ja sisältää standardipakettien lisäksi joitakin suositeltavia paketteja (ellei niitä ole jätetty pois käynnistyksen yhteydessä ladattavista). Sivustot ja käyttäjät voivat luoda omia paketteja ja käyttää niitä R-ohjelman ajon aikana. Funktiota .**libPaths**() voidaan käyttää polun määrittämiseksi hakemistoon, jossa kirjastot sijaitsevat, tai nykyisen sijainnin tulostamiseksi.

R-ohjelmisto käyttää automaattisesti sivustokohtaista kirjastoa **R_HOME/site − library**. Tämän sijaintia voidaan muuttaa profiilissa **R_HOME /etc/Rprofile.site** muuttamalla arvoa .**Library**.**site**. Kuten .**Library**, sivustokirjastot on aina sisällytetty hakemistopolkuihin .**libPaths**(). Jos oletuskirjasto on kirjoitussuojattu, sinne ei voi asentaa uusia paketteja (kts. Pakettien asentaminen). Oman kirjaston voi luoda **C : /myRlib** ja komennolla **library**(**pkgname**, **lib.loc** = "**C : /myRlib**") sinne ladataan asennetut paketit. Vaihtoehtoisesti voi ennen R-ohjelman käynnistämistä muuttaa ympäristömuuttujan arvoa, jolloin kirjaston sijaintia ei tarvitse joka kerta ilmoittaa erikseen kutsuttaessa funktiota **library**().

Käyttäjillä voi olla käytössään oma kirjasto tai niiden kokoelma, joka on eriteltynä ympäristömuuttujassa **R_LIBS_USER**. Ympäristömuuttujalla on oletusarvo, jonka näkee R-ohjelmiston ajon aikana komennolla **Sys.getenv**("**R_LIBS_USER**"). Käytössä olevien kirjastojen hakemistoja voi muuttaa ympäristömuuttujan arvoa muuttamalla, mikä tapahtuu kirjoittamalla "**R_LIBS_USER = C : /myRlibrary**" tiedostoon ".**Renviron**". Windowsia käytettäessä tämän voi tehdä ohjauspaneelista (esim. Windows 7:ssa): **System → Control Panel → User Accounts → Change my environment variables**.

Molemmat ympäristömuuttujat (**R_LIBS_USER** ja **R_LIBS_SITE**) voivat sisältää useita hakemistopolkuja kirjastoihin, kun ne erotetaan toisistaan kaksoispisteillä (Windowsissa puolipisteellä).

Käynnistyksen yhteydessä ladattavat paketit

Käynnistettäessä ladatut paketit ovat oletusarvoisesti

> **getOption**("**defaultPackages**")

[1] "*datasets*" "*utils*" "*grDevices*" "*graphics*" "*stats*" "*methods*"

ja näiden lisäksi R-standardin kirjasto **base**. Käynnistettäessä käyttöön otettavia kirjastoja voi muokata esimerkiksi profiilissa ∼ /.**Rprofile**. Käynnistyksen yhteydessä ladattavat paketit on tallennettu ympäristömuuttujaan **R_DEFAULT_PACKAGES** listana, jonka alkiot on erotettu toisistaan pilkuilla. Asettamalla **R_DEFAULT_PACKAGES = NULL** ladataan vain paketti **base**. Pakettien lataaminen jo käynnistysvaiheessa nopeuttaa R-ohjelmiston käyttöä.

Pakettien asentaminen

Unix

Paketteja jaetaan lähdekoodimuodossa tai käännettynä binäärikoodina. Lähdekoodipaketit, jotka sisältävät C, C++ tai Fortran -koodia, tarvitsevat vastaavan kääntäjän ja muut työkalut. Binääripaketit ovat alustakohtaisia (UNIX/Windows/OS X) ja asentuvat tavallisesti ilman erikoistyökaluja, mutta tämän voi tarkistaa pakettien dokumentaatiosta.

Mikäli käytössä on useampi kuin yksi kirjasto, on määriteltävä implisiittisesti (oletuskirjasto) tai eksplisiittisesti (kutsuparametri) kirjasto, jonne paketti asennetaan. Unixin kaltaisissa käyttöjärjestelmissä on

varmistettava, että järjestelmän **umask** on asetettu siten, että käyttäjällä on riittävät oikeudet pakettien asentamiseen.

Usein riittää kutsua funktiota **install.packages**(′**packagename**′) tai sen graafisen käyttöliittymän vastinetta, jos tarkoituksena on asentaa CRAN-jakelun paketti, ja käytettävissä on internet-yhteys. Useimmissa järjestelmissä funktiokutsun **install.packages**() jälkeen asennettava paketti on valittavissa aukeavasta listasta. Funktiolle voidaan paketin nimen lisäksi antaa muita parametreja. Tästä saa lisätietoa komennolla ?**install.packages**.

Unixin kaltaisissa käyttöjärjestelmissä pakettien asentaminen komentoriviltä (/**root**) tapahtuu komennolla

> **R CMD INSTALL** −**l /path/to/library packagename1 packagename2** ...

Osa −**l /path/to/library** voidaan jättää pois, jolloin paketit asentuvat ensimmäiseen käytettävään kirjastoon, joka on siis .**libPaths**()[**1**] eli listan ensimmäinen alkio. Aina on varmistettava, että ympäristömuuttuja **TMPDIR** on määrittelemättä (jolloin väliaikaisen hakemiston /**tmp** on oltava olemassa, kirjoitettavissa ja luettavissa) tai osoittaa validiin väliaikaishakemistoon. Asennuksen yhteydessä voi käyttää lukuisia lisävalitsimia, joista saa lisätietoa komennolla **R CMD INSTALL** −−**HELP**.

Vaihtoehtoisesti paketit voidaan ladata ja asentaa R-ohjelmiston ajon aikana. Aluksi on asetettava valitsin **CRAN**, eli käytettävä toisiopalvelin, mikä onnistuu funktiolla **chooseCRANmirror**(). Toisinaan käyttäjän on muokattava välityspalvelinasetuksiaan (tarkemmat ohjeet saa R-komennolla ?**download.file**). Useamman paketin kerralla voi ladata ja asentaa muodostamalla paketeista vektorin funktiolla **c**(). Jos paketit ovat esimerkiksi **packagename1** ja **packagename2**, niin ne voi ladata ja asentaa R-komennolla

> **install.packages**(**c**(′**packagename1**′, ′**packagename2**′))

Paketit ovat laajennuksia ja vaativat usein muita paketteja toimiakseen. Jos haluaa asentaa jonkin paketin ja kaikki ne paketit, jotka se vaatii toimiakseen, niin tämä onnistuu komennolla

> **install.packages**(′**packagename**′, **dependencies** = **TRUE**)

Ellei käytettyä kirjastoa, johon paketit asennetaan, ole määritetty kutsuparametrilla **lib**, käytetään ensimmäistä käytettävissä olevaa kirjastoa listasta .**libPaths**(). Jos kirjastoon ei ole kirjoitusoikeuksia R-ohjelmisto kysyy käyttäjältä oikeutta käyttäjän oletuskirjaston luomiseksi ja oikeutta asentaa paketit tähän kirjastoon.

Funktio **install.packages**() voi asentaa paketteja myös lokaalista .**tar.gz**-tiedostosta asettamalla komentoparametrin **repos** arvoksi **NULL** (Tämä tapahtuu automaattisesti, jos annettu paketinnimi on yksittäinen .**tar.gz** tiedosto).

Funktio **install.packages**() käy läpi useita pakettivarastoja, jos komentoparametrin **repos** arvoksi annetaan vektori, johon voidaan sisällyttää mm. CRAN mirror, Bioconductor, Omegahat, R-forge, lokaalit arkistot, lokaalit tiedostot.... Funktiolla

> **setRepositories**()

voidaan asettaa R-ohjelmistolle näkyvät pakettivarastot.

Käyttäjän on muistettava ottaa asentamansa paketti, eli kirjaston toiminnallisuus, käyttöön komennolla

> **library**(**packagename**)

Windows

Pakettien asentaminen toimii periaatteessa samoin kuin Unixilla, mutta se, mitä funktio **install.packages**() tekee, eroaa oletusarvoisesti Windowsilla. Unixilla funktio käy läpi *lähdekoodi*paketit määritellyistä pakettivarastoista (CRAN, lokaalit tiedostot,...), lataa viimeisimmän version ja asentaa sen. Windowsilla funktiota kutsuttaessa se käy (oletusarvoisesti) läpi saatavilla olevat *binääri*paketit ja asentaa viimeisimmän version, jos mahdollista. Funktiolla voi ladata ja asentaa myös *lähdekoodi*paketin, mutta tästä on ilmoitettava sille kutsuparametrilla seuraavasti

> **install.packages**(′**packagename**′, **type** = ′**source**′).

Windowsilla funktiolla **install.packages**() voi asentaa binääripaketteja myös lokaalista **zip**-tiedostosta asettamalla funktiokutsun **repos**-argumentin arvoksi **NULL**.

Helpoiten pakettien lataaminen käy lataamalla heti aluksi graafinen käyttöliittymä: esimerkiksi Windows-asennustiedoston mukana tulevassa käyttöliittymässä **Rgui.exe** on valikko **Packages**, josta löytyvät toiminnot **setRepositories**(), **install.packages**() ja **update.packages**().

Muutamien binääripakettien asentaminen vaatii ulkoista ohjelmistoa. Jos ollaan asentamassa 64-bittiselle Windowsille paketteja **Cairo**, **RGtk2**, **cairoDevice** tai näistä riippuvaisia paketteja, niin kannatta katsoa linkki

`http://www.gtk.org/download-windows-64bit.html`.

Lähdekoodipakettien asentaminen ei vaadi ulkoisia ohjelmistoja, kunhan paketit eivät sisällä käännettyä koodia. Tyypillisesti Windowsille saatavat paketit ovat binäärimuotoisia ja niiden asentaminen suoraviivaista, mutta mikäli pakettien kokoaminen suoraa lähdekooditiedostosta on välttämätöntä ja ne sattuvat sisältämään käännettyä koodia, niin lista tarvittavista työkaluista ja niiden käyttöohjeet löytyvät osoitteesta

`http://cran.r-project.org/doc/manuals/R-admin.html`, (`Appendix D`).

Lähdekoodipakettien kokoamisen jälkeisiä satunnaisia ongelmia on raportoitu Vista/Windows/Server2008 alustoilla, ja ongelmat on voitu kiertää asettamalla ympäristömuuttujan **circumvented** arvoksi **tar.exe**.

Jos haluaa muuntaa tiettävästi toimivan lähdekoodipaketin Windowsin binääritiedostoksi, voi käyttää esimerkiksi R-projektin tarjoamaa muuntajaa

`http://win-builder.r-project.org/`.

OS X

Pakettien asennus käy samoin kuin muilla Unixin kaltaisilla käyttöjärjestelmillä, mutta saatavilla on erillinen binäärijakelu **mac.binary**$*$, joka voidaan ladata ja asentaa funktiolla **install.packages**(). Esimerkiksi suurimman pakettivaraston CRAN jakelut ovat **mac.binary.leopard** (tiedostot **.tgz**).

Ohjelman R.app graafisen käyttöliittymän valikosta on valittavissa asennetaanko binääri- vai lähdekoodipaketti ja ladataanko se CRAN:sta vai lookalista hakemistosta.

Useat aliarkkitehtuurit

Lähdekoodipaketteja asennettaessa on hyvä muistaa, että asennus tapahtuu aina jollekin arkkitehtuurille. Näihin törmää usein OS X:n ja Windowsin tapauksissa (32bit/64bit), mutta myös muilla alustoilla.

Kun R asentaa lähdekoodipakettia, joka tukee useita aliarkkitehtuureja, niin tavallisesti paketit asentuvat tukemaan kaikkia käytettävissä olevia aliarkkitehtuureita, mutta asennusvaiheessa testataan vain käynnissä olevan R-ohjelmiston arkkitehtuuria vastaava versio. Poikkeuksia ovat paketit

Unixille, joissa on **configure**-skripti tai tiedosto **src/Makefile**.

Windowsille, joissa on **configure.win**-skripti tai tiedosto **src/Makefile.win**.

Jos **configure.win** on arkkitehtuurista riippumaton, niin paketti asentuu tukemaan kaikkia käytettäviä aliarkkitehtuureja. Asennusohjelma voidaan myös pakottaa asentamaan paketti kaikkia aliarkkitehtuureja tukevaksi lisävalitsimella –**force-biarch**.

Pakettien päivittäminen

Paketteja asennettaessa on aina hyvä varmistaa, että paketit, joista asennettavat paketit riippuvat, eli joiden toiminnallisuutta ne sisältävät, ovat ajan tasalla. Helpoiten tämä käy komennolla

> **update.packages**()

Funktiolle annettava argumentti **repos** määrittelee, mistä pakettivarastoista funktio tarkastaa pakettien viimeisimmän version. Funktio lataa listan paketeista, vertaa listaa käyttäjän asentamiin paketteihin ja päivittää vanhentuneet paketit.

Funktiolla **packageStatus**() voi tarkastaa asennettujen pakettien tilan. Funktio palauttaa listan kirjastoistasi ja niiden sisältämien pakettien tilan: "**ok**", "**upgrade**" tai "**unavailable**".

Pakettien poistaminen

Paketteja ***voi*** poistaa komentoriviltä käsin komennolla

>R CMD REMOVE -l /path/to/library pkgname1 pkgname2 ...

tai R-ohjelman ajon aikana komennolla

> remove.packages(c("pkgname1", "pkgname2"),lib = file.path("path", "to", "library"))

tai suoraan kansiosta.

Datan lukeminen

R-ohjelmiston valmiiden datanlukutyökalujen lisäksi on R-ohjelmistolle tehty paketteja, joiden avulla voidaan tuoda dataa muista ohjelmistoista ja järjestelmistä. Pakettien käyttöönotto-ohjeet löytyvät luvusta *Pakettien asentaminen*.

Vektori- ja ASCII-data

Funktiolla **scan**() voidaan lukea dataa näppäimistöltä ja vektorimuotoista dataa **.dat**-tiedostosta

$$> \mathbf{myData} <- \mathbf{scan}(\text{"}\mathbf{myData.dat}\text{"})$$

Jättämällä edellisestä pois parametrin "**myData.dat**", funktio lukee dataa näppäimistöltä. Datan syöttämisen voi lopettaa painamalla kahdesti näppäintä Enter.

Funktiolla **read.table**() voi lukea ASCII-muotoista dataa, esimerkiksi **.txt**-tiedostosta. Funktiolle voidaan antaa lisäparametreja esimerkiksi kirjoittamalla

$$> \mathbf{myData} <- \mathbf{read.table}(\text{"}\mathbf{C:/path/to/myData.txt}\text{"}, \mathbf{sep} = \text{","}, \mathbf{header} = \mathbf{TRUE}).$$

Lisäparametrin **sep** avulla voidaan lukea dataa, jonka alkiot on erotettu pilkulla, ja parametri **header** jättää otsikkorivin lukematta. Tarkempaa tietoa lisäparametreista saa komennolla ?**read.table**().

Excel-data

Helpoin tapa tuoda dataa Excelistä on tallentaa tiedostot **.csv**-formaattiin, jolloin niitä voi lukea samoin kuin **.txt**-tiedostoja (kts. edellinen kappale). Excelin **.xls**-tiedostojen lukemiseksi tarvitaan pakettia **RODBC** (Unixilla **unixODBC**). Excel-datan luku on kolmivaiheinen. Aluksi avataan kanava komennolla

$$> \mathbf{channel} <- \mathbf{odbcConnectExcel}(\text{"}\mathbf{C:/path/to/myExelData.xls}\text{"})$$

Tämän jälkeen tallennetaan data muuttujaan **myData**

$$> \mathbf{myData} <- \mathbf{sqlFetch}(\mathbf{channel}, \text{"}\mathbf{myExcelSheet}\text{"})$$

Lopuksi suljetaan kanava komennolla

$$> \mathbf{odbcClose}(\mathbf{channel})$$

SPSS- ja SAS-data

SPSS- ja SAS-ohjelmistoissa on data ennen lukemista tallennettava siirrettävään muotoon. Nämä tiedostot voidaan lukea R-ohjelmistoon paketilla **Hmisc**. SPSS-ohjelmiston siirtoformaatti on **.por** ja se luetaan R-ohjelmistoon komennolla

$$> \mathbf{myData} <- \mathbf{spss.get}(\text{"}\mathbf{C:/myData.por}\text{"}, \mathbf{use.values.labels} = \mathbf{TRUE})$$

Vastaavasti SAS-ohjelmiston siirtoformaatti on **.xpt**, ja se luetaan komennolla

$$> \mathbf{myData} <- \mathbf{sasxport.get}(\text{"}\mathbf{C:/myData.xpt}\text{"})$$

Stata- ja Systat-data

Ottamalla käyttöön kirjasto **foreign** voidaan Stata- ja Systat-ohjelmistojen datatiedostot (**.dta**) lukea R-ohjelmistoon suoraan komennoilla

$$> \mathbf{myData} <- \mathbf{read.dta}(\text{"}\mathbf{C:/myData.dta}\text{"}) \ \#\textit{Statasta}$$

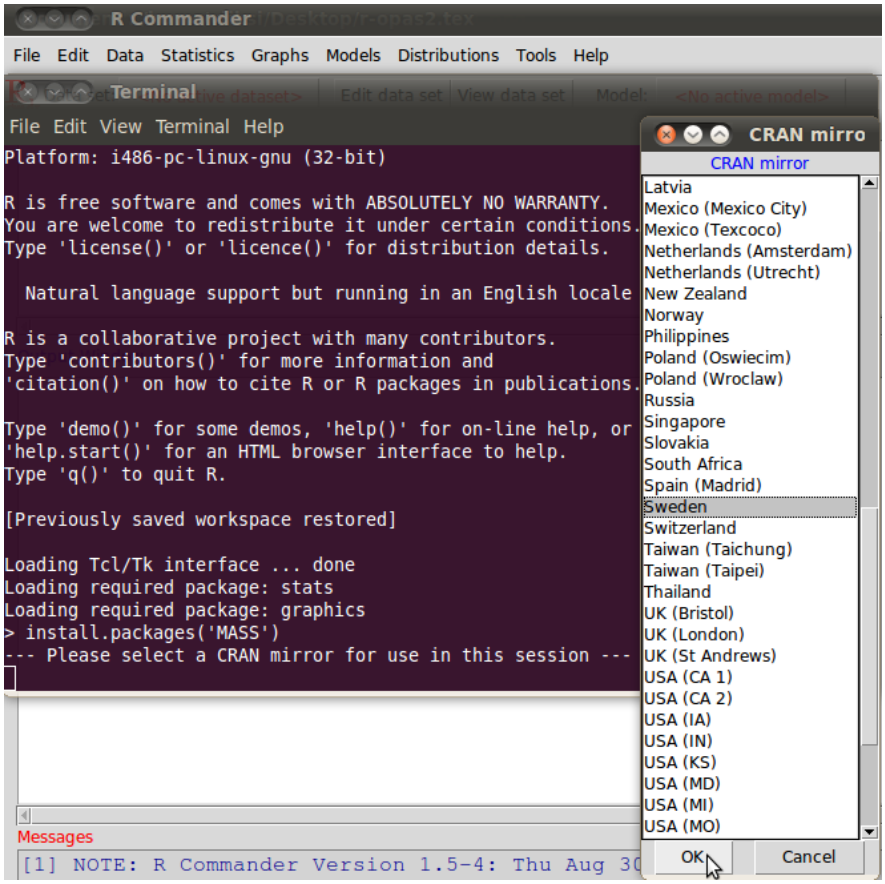
ja

$$> \mathbf{myData} <- \mathbf{read.systat}(\text{"}\mathbf{C:/myData.dta}\text{"}) \ \#\textit{Systatista}$$

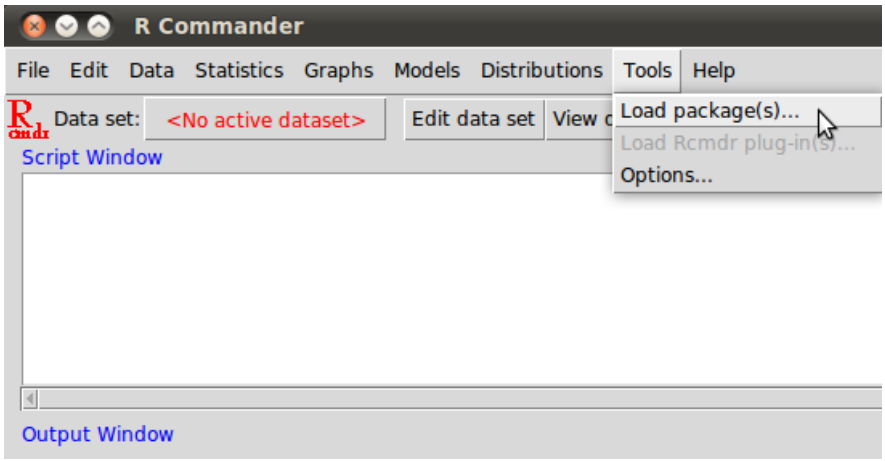
Esimerkit

Kirjaston käyttöönotto

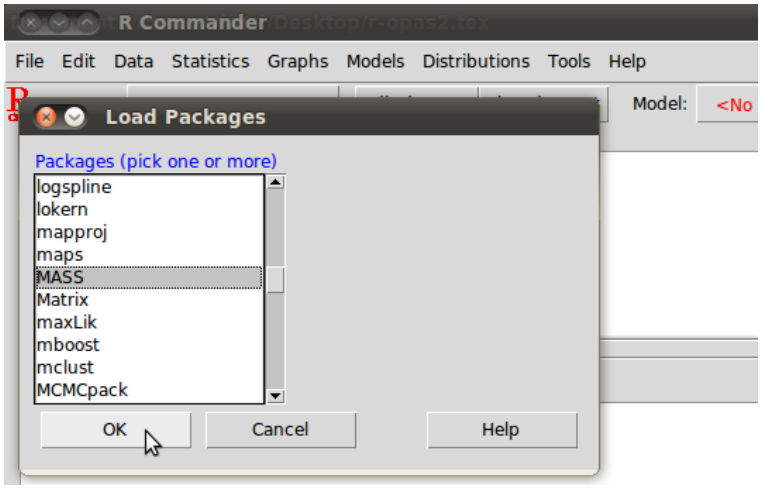
Esimerkissä otetaan käyttöön kirjasto **MASS**. Käyttöönotto aloitetaan kirjoittamalla komentoriville asennuskomento **install.packages**('**MASS**') ja tämän jälkeen valitaan käytettävä palvelin.



Palataan nyt R-ohjelman graafiseen käyttöliittymään (R Commanderiin), jolloin paketin voi ottaa käyttöön.



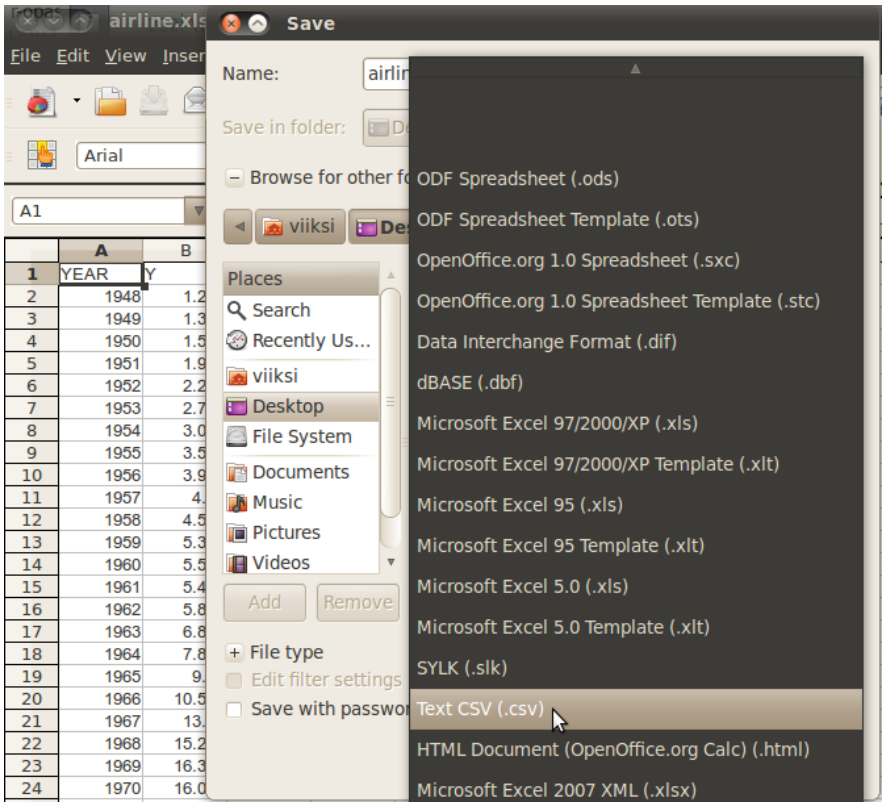
Etsitään seuraavaksi asennettu paketti listasta ja ladataan se klikkaamalla OK.



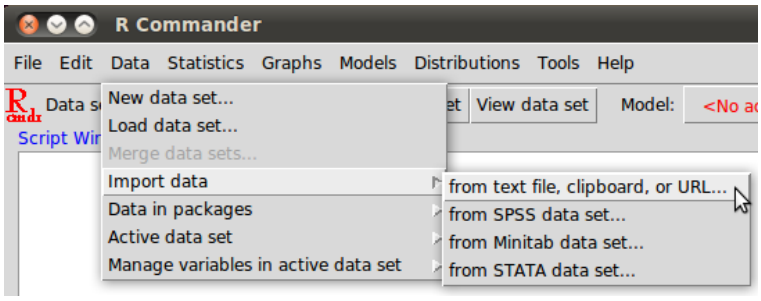
Paketin toiminnallisuus on nyt käytettävissäsi kunnes R-ohjelman suljetaan.

Datan lukeminen

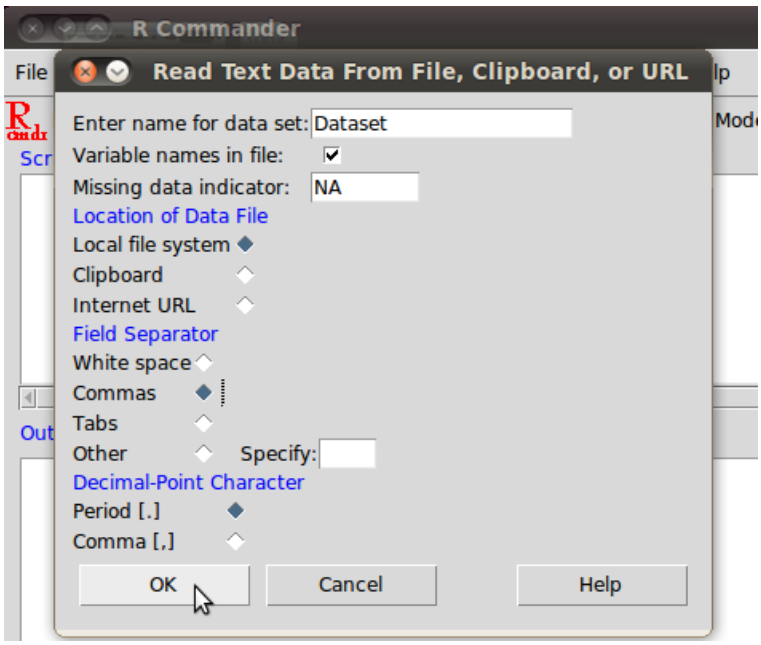
Esimerkissä käytetty **airline.xls**-tiedosto on ladattu osoitteesta
`http://www.principlesofeconometrics.com/excel.htm`



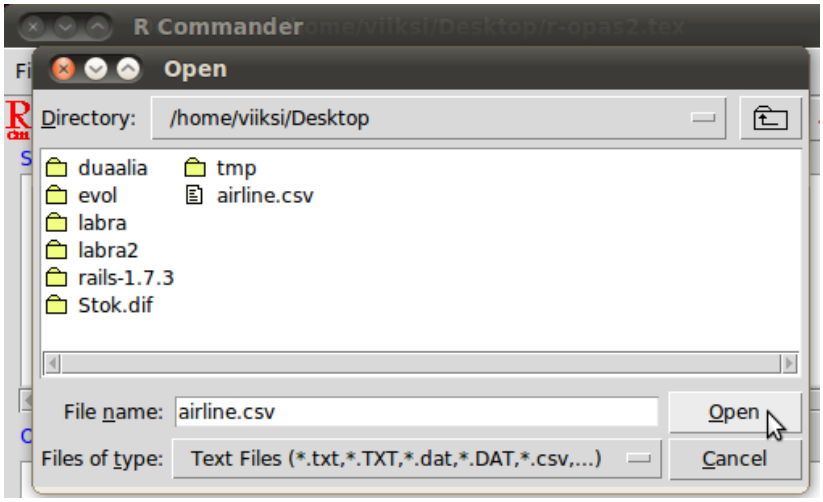
Avataan tiedosto ja tallennetaan se **.csv**-tiedostona. Nyt voidaan palata R-ohjelmaan ja lukea data tiedostosta.



Valitaan valikkopaneelista **Data** → **Import data** → **from text file, clipboard, or URL**...



Nimetään muuttuja, johon data tallennetaan (esimerkissä **Dataset**). Lisäksi ohjelmalle on ilmoitettava missä luettava **.csv**-tiedosto sijaitsee (esimerkissä kovalevyltä), alkiot erottava merkki (esimerkissä pilkku) ja desimaalimerkki (esimerkissä piste).



Etsitään tiedosto valikkohakemistosta. Nyt data on luettu muuttujaan **Dataset**.