Introduction to Item Response Theory

Kathleen Preston, Ph.D.

Western Psychological Association
Portland, OR


IRT TO THE PEOPLE

Outline

- Review Measurement and Classical Test Theory
- Establish Item Response Theory Fundamentals
- Compare Classical Test Theory and Item Response Theory
- Assumptions
- Dichotomous Models
- Example Using R
- Interpretation

Measurement

- Definitions for consideration
  - Assigning numbers to individuals in a systematic way
  - Applied to the properties of objects rather than to the objects themselves
- Conceptualize the latent variable or construct
  - Products of the informed scientific imagination of social scientists who attempt to develop theories for explaining human behavior
  - Represents what is common across manifested behaviors
  - The hidden source of common variability or covariability of a set of similar observable behaviors
  - Psychological theory is a statement about a possible relationship between two or more psychological constructs or between a construct and an observable behavior
- Establish the operational definition
  - What are the behavioral manifestations of the construct?

Classical Test Theory

- Evaluates the extent to which observed scores are affected by measurement error
- Classical Test Theory decomposition

$$X = T + E$$

  - True score: average of all measurements obtained after a test is administered many times to the same individual ensuring measurements are i.i.d (independent and identically distributed).

$$T_{ki} = \bar{X}_{ki}$$

    - Test and subject specific

  - Error score: the complement of an individual's true score to the observed score on a given test

$$E = X - T$$

IRT Origins

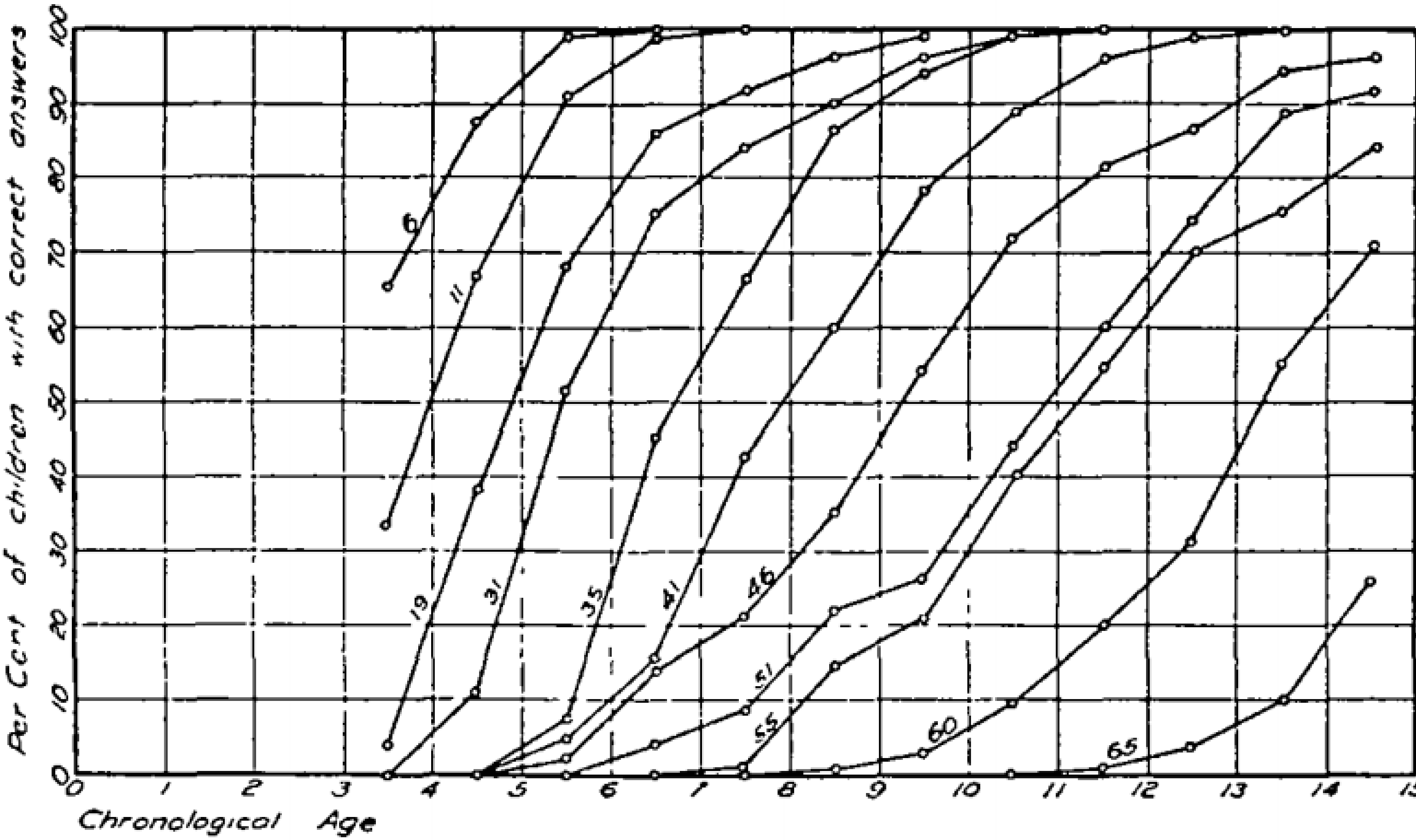


FIG. 5.

Proportions of correct response to selected items from the Binet-Simon test among children in successive age groups (Thurstone, 1925)

IRT Origins

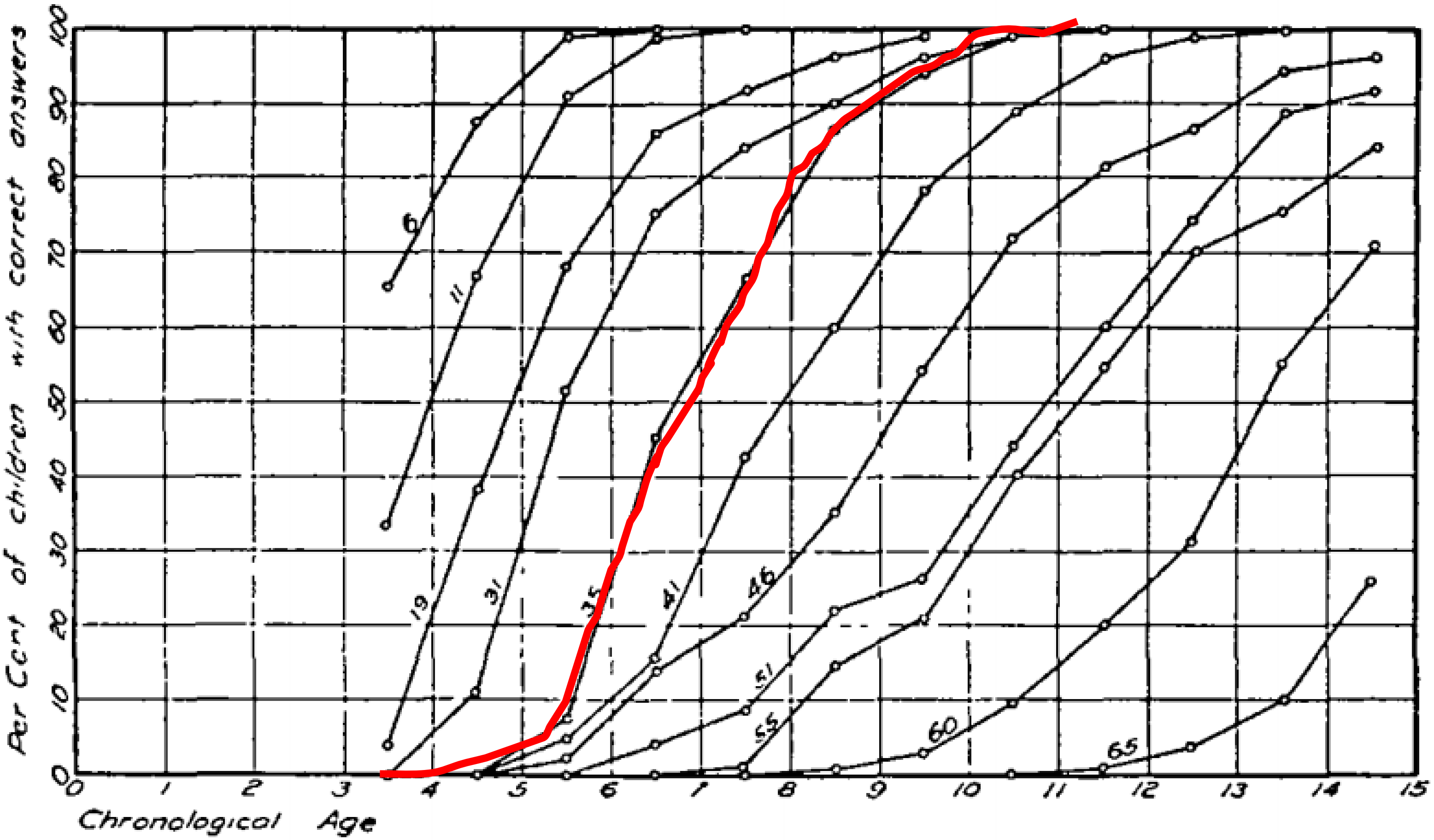
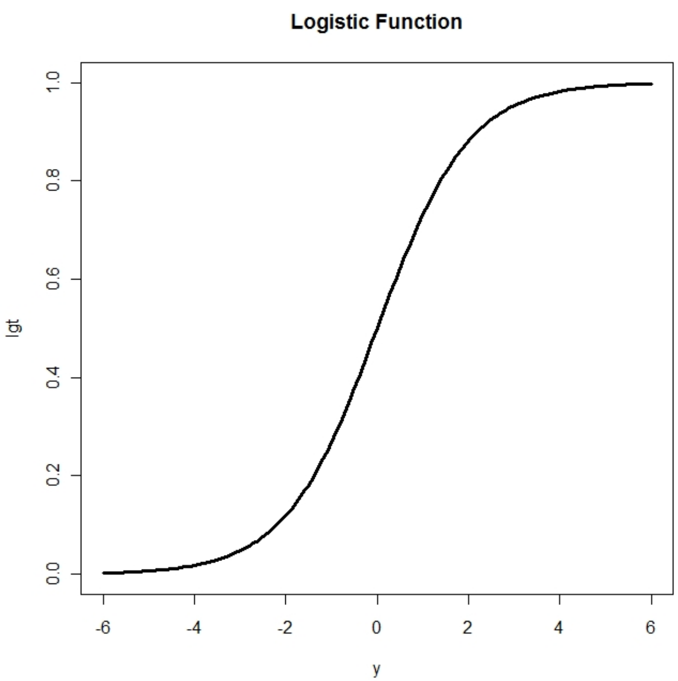


FIG. 5.

Proportions of correct response to selected items from the Binet-Simon test among children in successive age groups (Thurstone, 1925)

Logistic Curve

$$f(z) = \frac{1}{1 + e^{-z}}$$

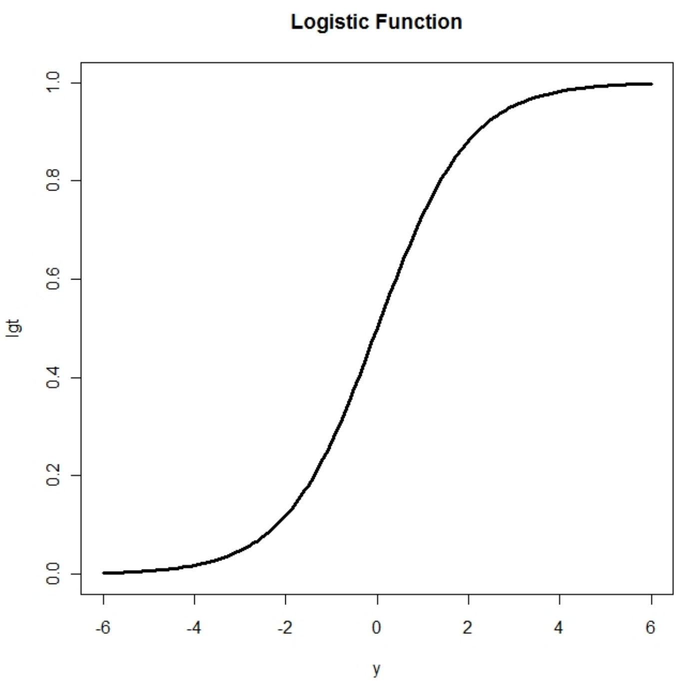


Logistic Curve

$$f(z) = \frac{1}{1 + e^{-z}}$$

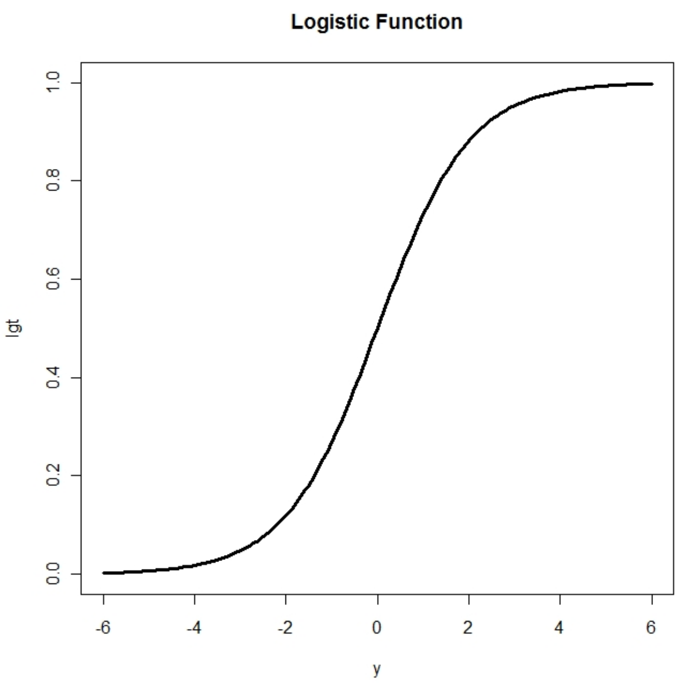
Logistic Regression

$$z = \beta_0 + \beta_1 x_1$$



Logistic Curve

$$f(z) = \frac{1}{1 + e^{-z}}$$



Logistic Regression

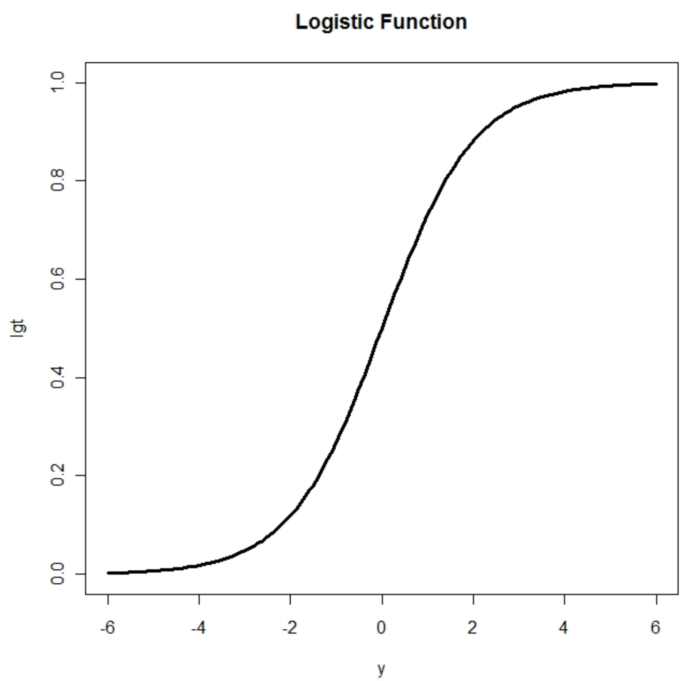
$$z = \beta_0 + \beta_1 x_1$$

Item Response Theory

$$z = a\theta + b$$

Logistic Curve

$$f(z) = \frac{1}{1 + e^{-z}}$$



Logistic Regression

$$z = \beta_0 + \beta_1 x_1$$

Item Response Theory

$$z = a\theta + b$$

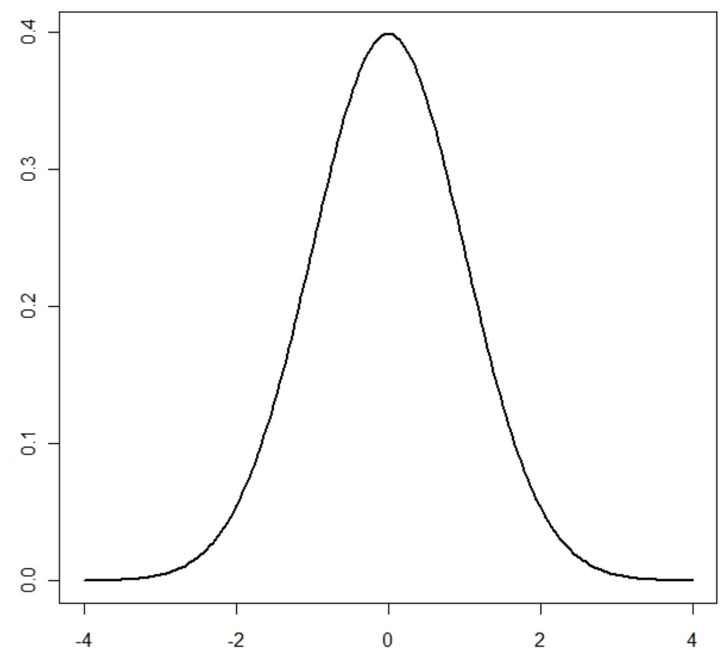
$$P_j(\theta) = \frac{1}{1 + e^{-(a_j\theta + b_j)}}$$

$$a_j = \text{item discrimination}$$

$$b_j = \text{item location}$$

θ

- Represents individual differences on construct or “latent trait” being measured
- The human capacity measured by the test
  - i.e. knowledge level, cognitive ability, personality characteristic
- Z-score interpretation



IRT Terms

- **Assumption of local independence**: a response to a question is independent of responses to other questions in a scale after controlling for the latent trait measured by the scale

- **Assumption of unidimensionality**: the set of questions are measuring a single continuous latent variable

- **Information function (IIF/TIF)**: an index indicating the range of trait level over which an item or test is most useful for distinguishing among individuals
  - Characterizes the precision of measurement for persons at different levels of the underlying latent construct, with higher information denoting more precision

- **Item**: question on a scale

More IRT Terms

- **Item Characteristic Curve (ICC)**: The ICC models the relationship between a person's probability for endorsing an item category and the level on the construct measured by the scale

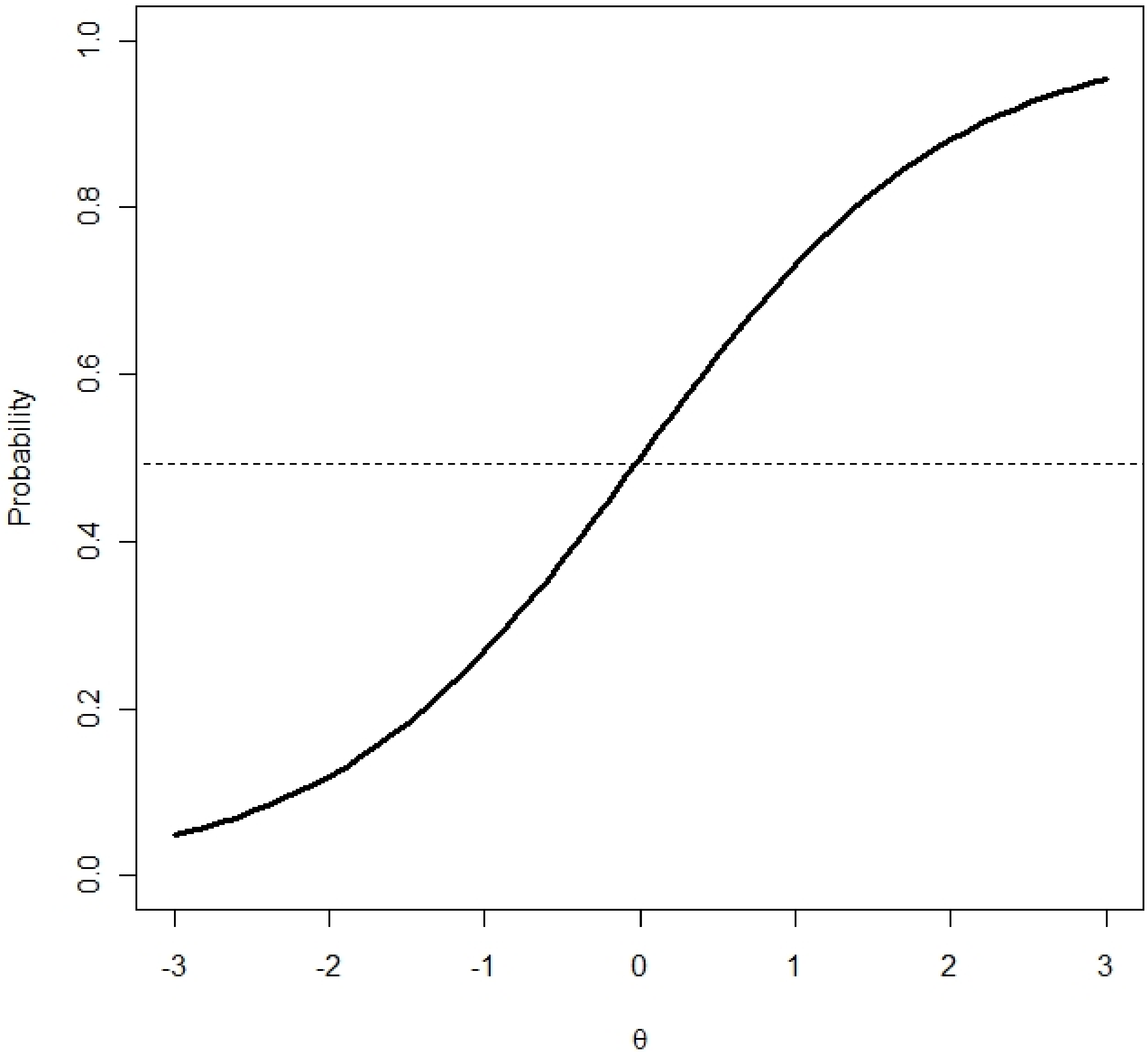
- **Item Difficulty (threshold) parameter *b***: the point on the latent scale where a person has a 50% chance of responding positively to the scale item

- **Item Discrimination (slope) parameter *a***: describes the strength of an item's discrimination between people with trait levels below and above the threshold *b*.
  - The *a* parameter may also be interpreted as describing how an item may be related to the trait measured by the scale

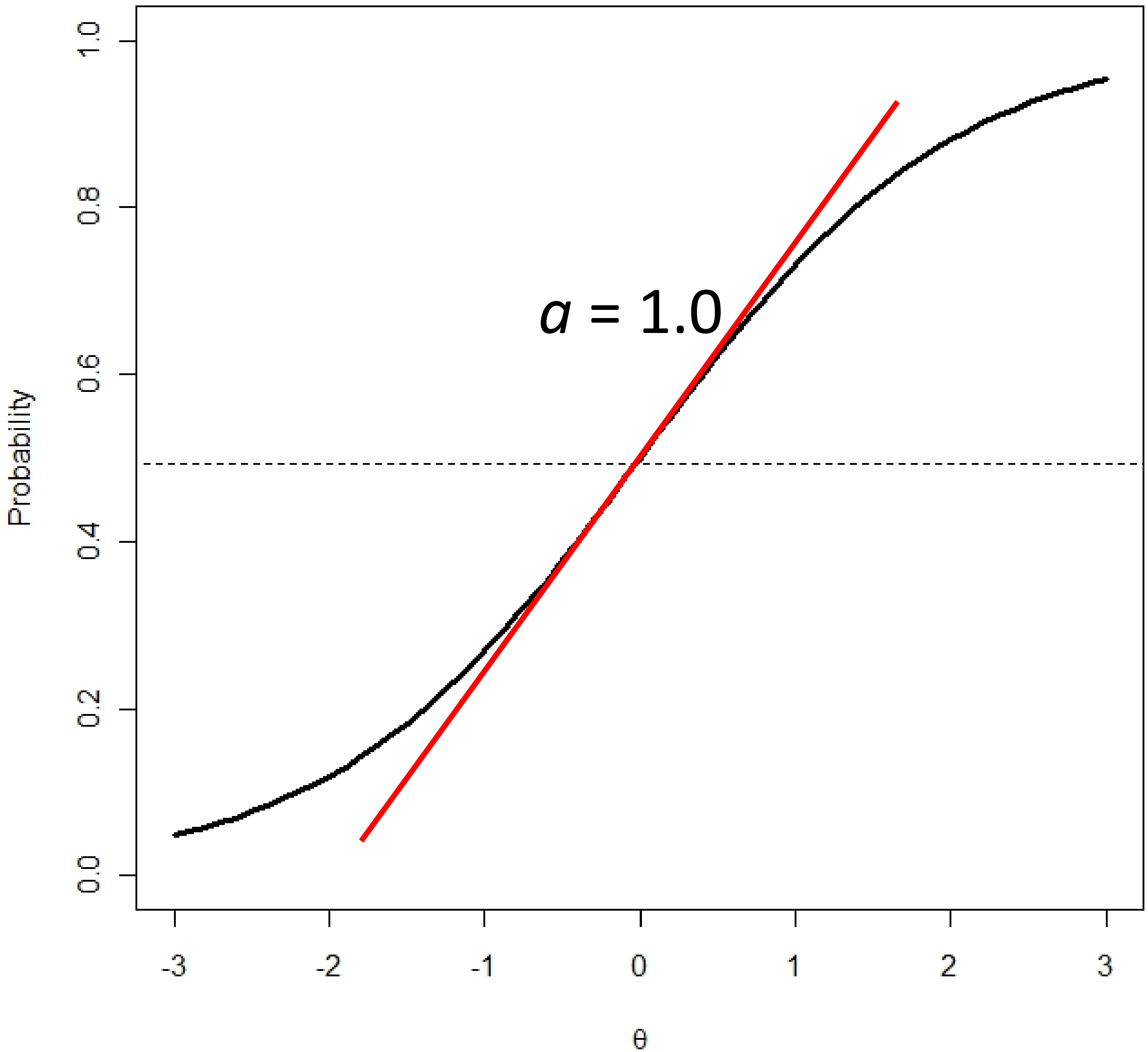
- **Test Characteristic Curve (TCC):** describes the expected number of scale items endorsed as a function of the underlying latent variable

- **Theta (θ):** the unobservable construct being measured by the questionnaire

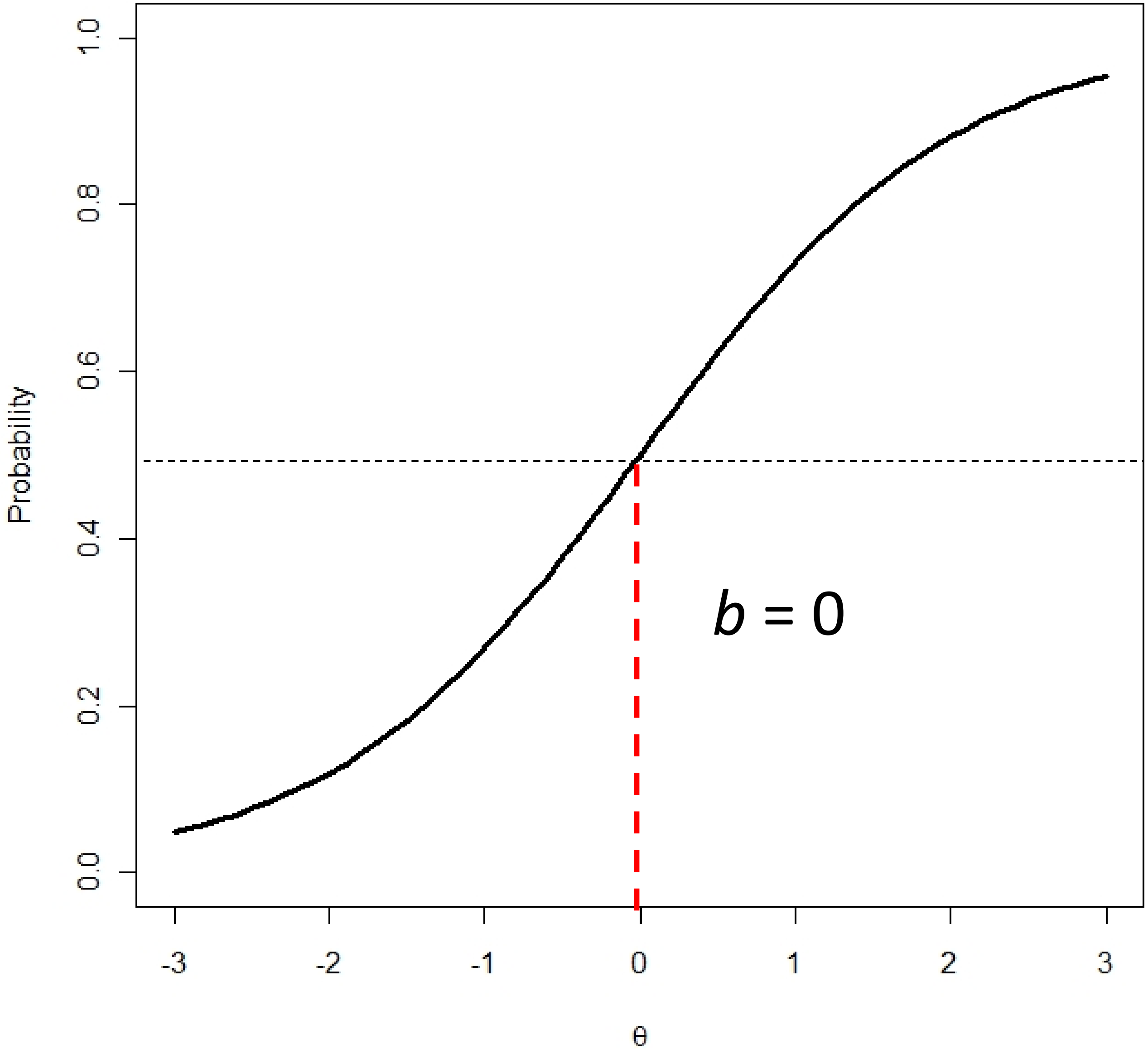
Item Characteristic Curve



Item Characteristic Curve

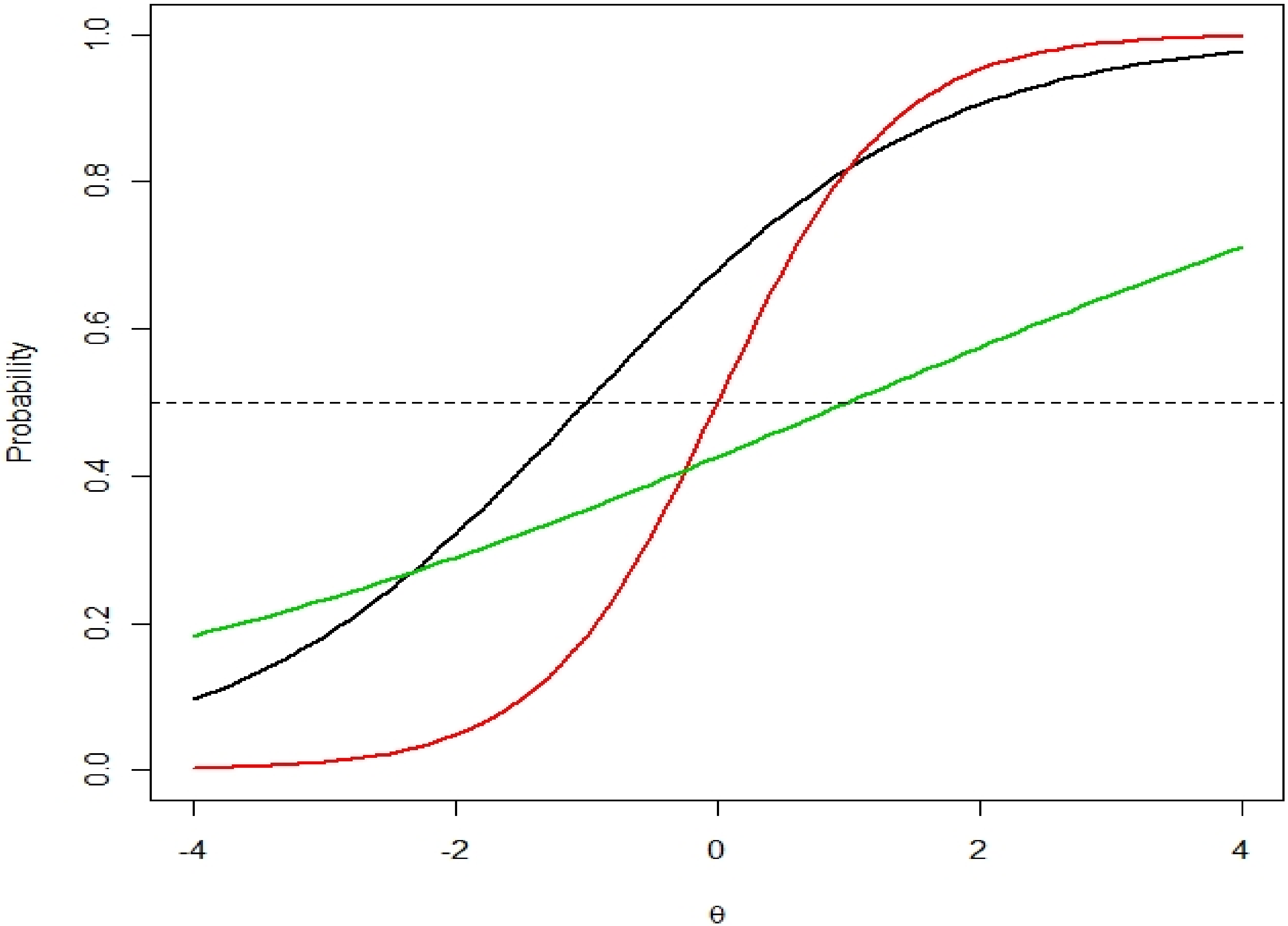


Item Characteristic Curve



Item Characteristic Curves

a = 0.75, 1.5, 0.3 b = -1, 0, 1



CTT versus IRT

Differences between CTT and IRT

- Test Precision
  – Reliability

$$\alpha = \frac{p}{p-1}\left[1 - \frac{\sum_{k=1}^{p} \sigma_{X_k}^2}{\sigma_X^2}\right]$$

  - α = .97

  – Information

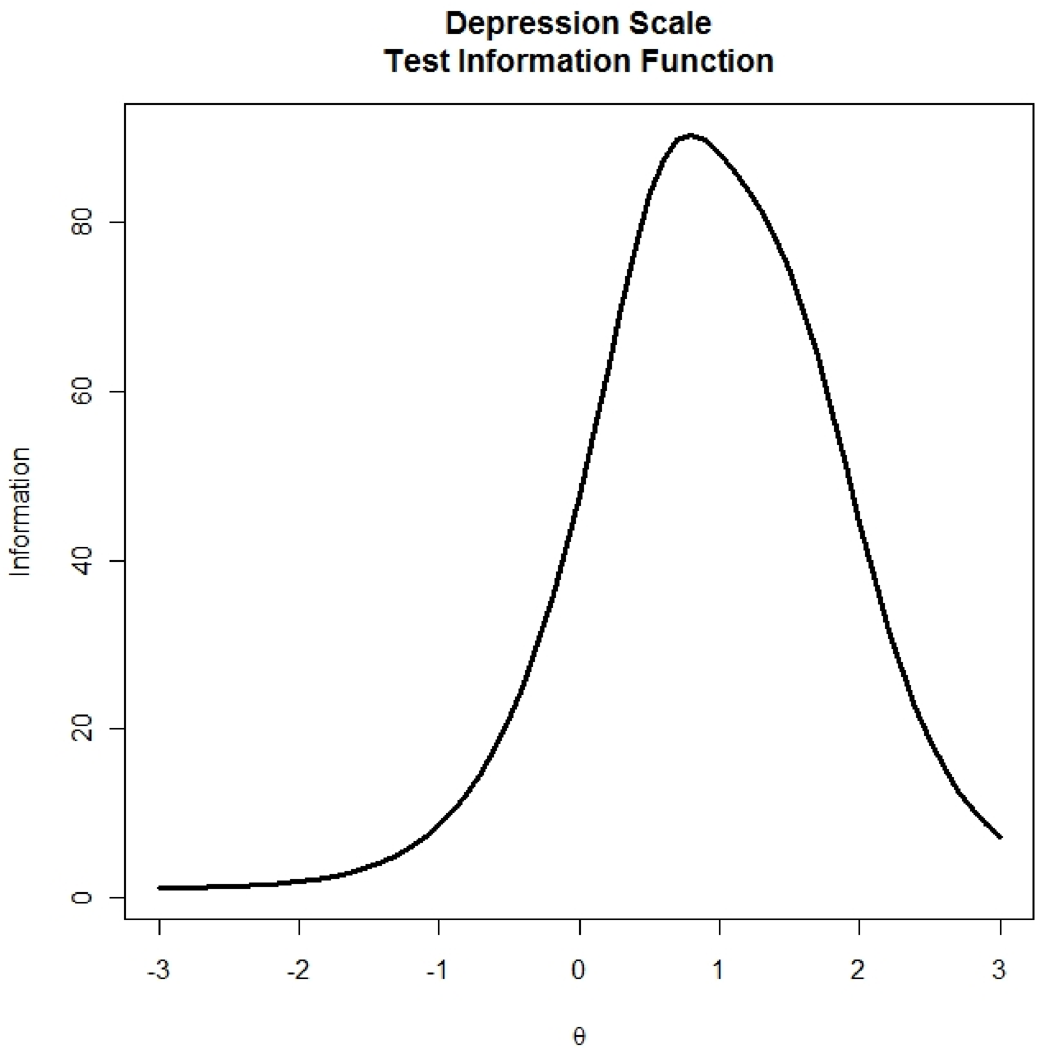
  •For 1- and 2-PL

$$I(\theta) = a_i^2 P_i(\theta)(1 - P_i(\theta))$$

$$TIF(\theta) = \sum_{i=1}^{I} I(\theta)$$

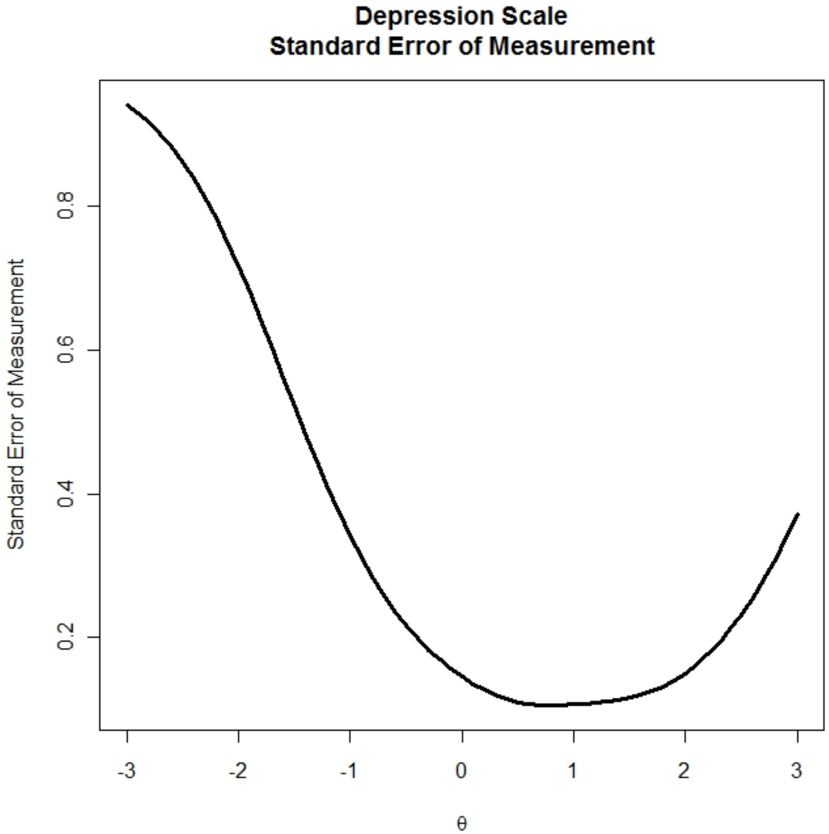
– Composite reliability

$$r_{tt} = \frac{\sigma_\theta^2}{\sigma^2}$$



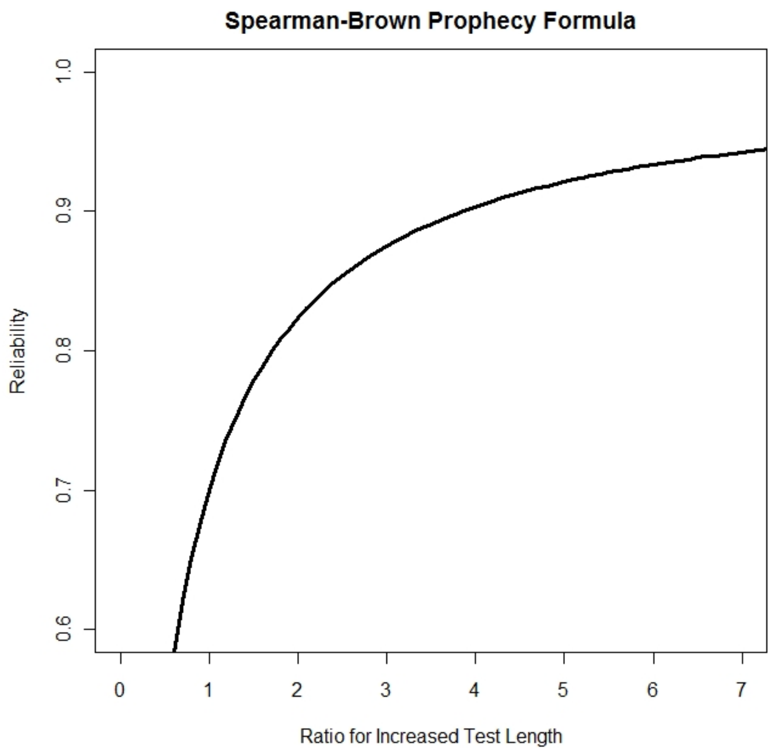
Differences between CTT and IRT

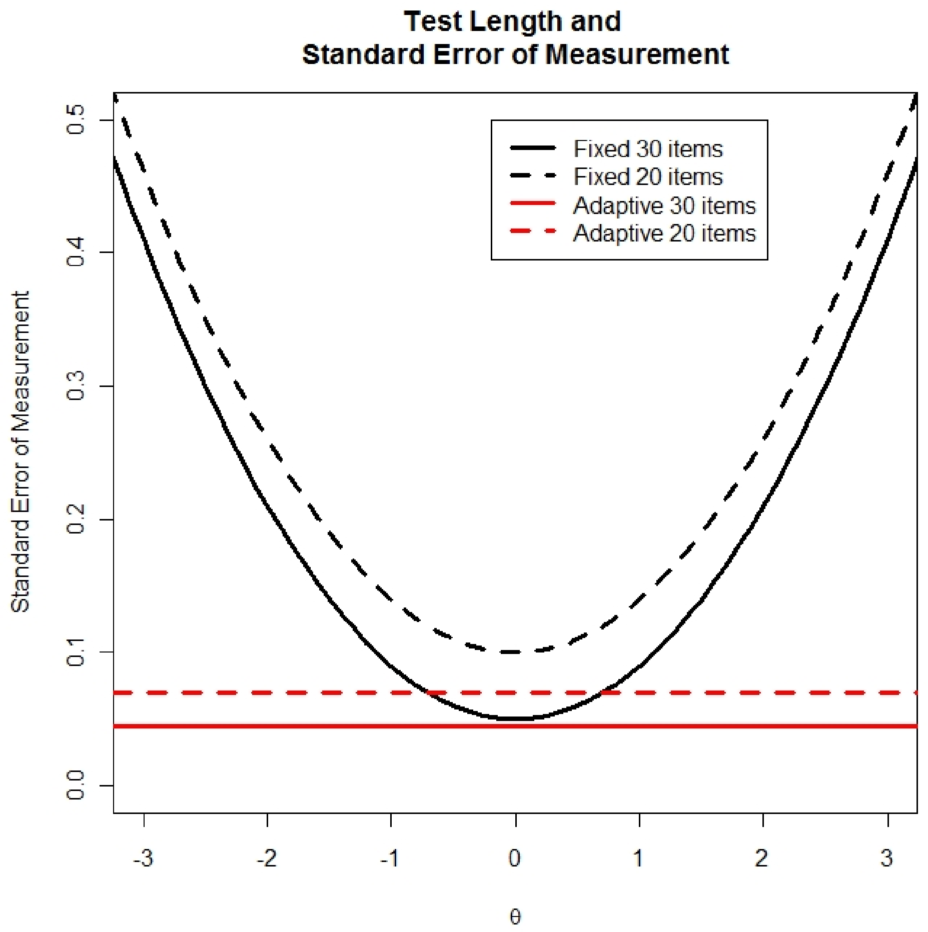
- Errors of Measurement
  - Constant for all scores
    - $SEM = \sqrt{(1 - r_{tt})}\sigma$
    - SEM = 3.08
  - Depends on trait level
    - $SEM|\theta = \dfrac{1}{\sqrt{TIF|\theta}}$
    - Composite error
      - $\sigma_\theta = \dfrac{SEM|\theta}{\theta}$



Differences between CTT and IRT

- Test Length
  - Longer tests are more reliable
    - $\rho = \frac{k\rho}{1+(k-1)\rho}$
  - Shorter tests can be more reliable



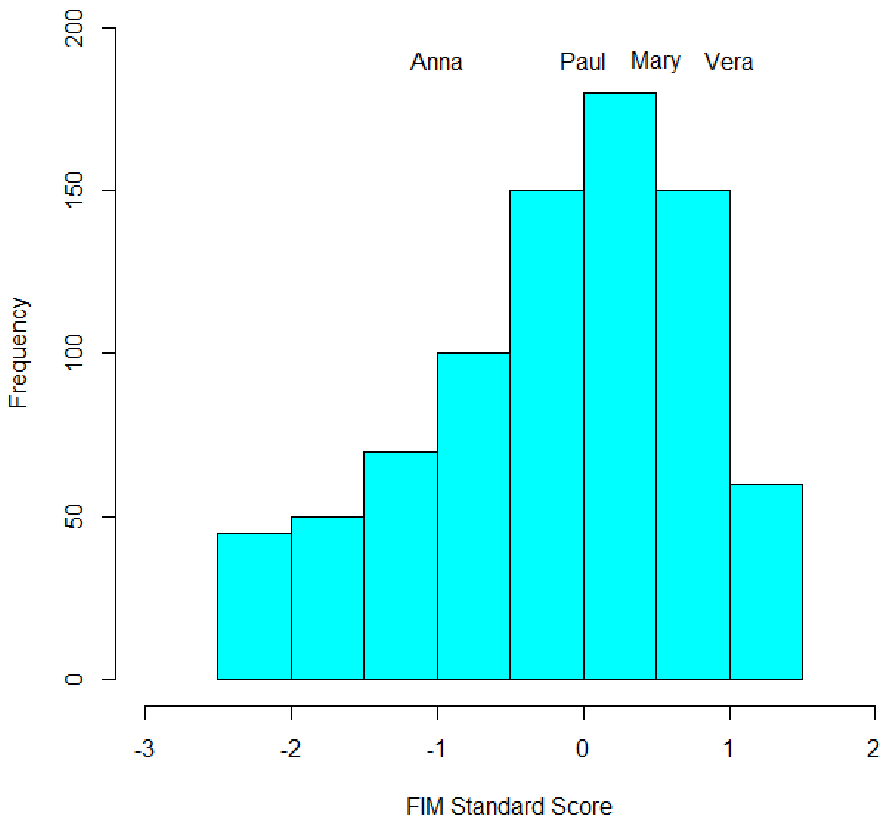


Differences between CTT and IRT

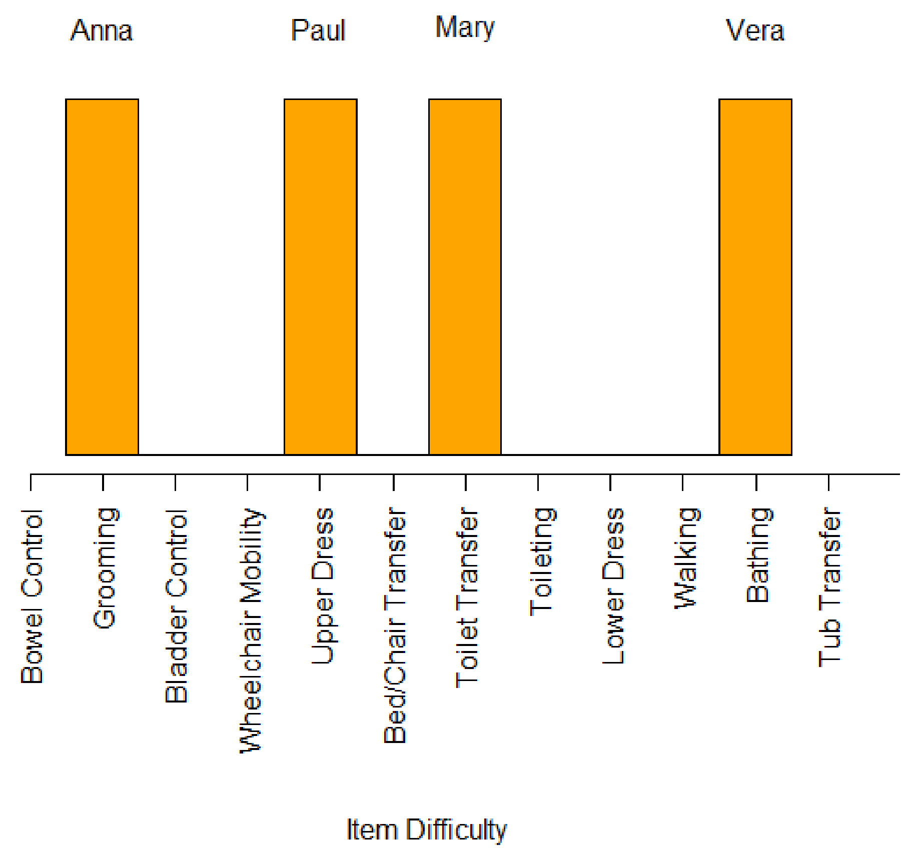
- Interchangeable test forms
  - Tests must be parallel to compare across multiple forms
  - Comparing multiple forms is optimal when difficult levels vary between persons
- Unbiased assessment of item properties
  - Requires representative sample
  - May be obtained from unrepresentative samples
- Change scores
  - Cannot be meaningfully compared when initial score levels differ
  - Can be meaningfully compared when initial score levels differ

Differences between CTT and IRT

- Compare individuals' position in a norm group



- Compare individuals' distance from items



Differences between CTT and IRT

- Different Response Formats
  - Problem
  - No Problem
- Test properties
  - Sample dependent
  - Sample free – given that the model fits that data
- Scoring Individuals
  - Raw Scores – summed scores are on an ordinal scale
  - Trait Level Estimates – theta scores are on an interval scale
- Factor analysis of dichotomous items
  - Produces artifacts rather than factors – creates difficult factors from phi correlations
  - Yields full information factor analysis

IRT Assumptions

- Unidimensionality: items are measuring a single continuous latent variable θ ranging from $-\infty$ to $+\infty$
  - Evaluate dimensionality using ECV index (Reise, 2012)
    1. Estimate bifactor model from poly/tetrachoric correlations
    2. Apply a Schmid-Leiman (SL) transformation to the factor loadings to obtain a semi-restricted exploratory hierarchical solution
    3. ECV_2 - Ratio of variance explained by the general factor to variable explained by general and group factors

$$ECV_2 = \frac{\sum(\lambda_G)^2}{\sum(\lambda_G)^2 + \sum\left(\lambda_{g1}\right)^2 + \sum\left(\lambda_{g2}\right)^2 + \sum\left(\lambda_{g3}\right)^2}$$

  - Violation
    - Multidimensional IRT
    - Separate IRT models for each subscale (dimension)
- Local Independence: item responses are independent of one another
  - Evaluate using Yen's (1984) Q3 correlations
    - Evaluate further if Yen's Q3 correlations diverge from the expected value of $-1/(n-1)$, where *n* is the number of items
  - Violation
    - Testlet: combine two dependent items into an item with ordered levels

Dichotomous IRT Models

- Models the relationship between an unobserved variable and the probability of the examinee correctly responding to any particular item
- Common dichotomous IRT models
  - 1PL or Rasch model
  - 2PL
  - 3PL

2 Theories toward measurement

1PL

- Goal is to develop a well-fitting model to reflect the data by parameterizing the ability of interest as well as the properties of the items
- The model should reflect the properties of the data
- Items are assumed to measure as they do, not as they should
- Measurement is to explain the data
- Only requires the slope to be equal for all items

Rasch

- The data must fit the specific measurement properties defined by the model
- If the item or person does not fit, then they are discarded
- Believes optimal measurement is defined mathematically
- *Specific objectivity:* comparison of two items' difficulty are assumed to be independent; and subjects' trait levels are not dependent on any subset of items administered
- Slope is fixed at 1 for all items

Common IRT models

1PL – items can vary in difficulty

$$P(\theta) = \frac{1}{1 + \exp[-a(\theta - b_i)]}$$

2PL – items can vary in discrimination and difficulty

$$P(\theta) = \frac{1}{1 + \exp[-a_i(\theta - b_i)]}$$

3PL – items can vary in discrimination, difficulty, and guessing (respondents may over report desirable, or underreport embarrassing behaviors)

$$P(\theta) = c_i + \frac{1 - c_i}{1 + \exp[-a_i(\theta - b_i)]}$$

Example:
Rosenberg Self-Esteem Scale (1965)

1. On the whole, I am satisfied with myself.
2. At times, I think I am not good at all (R)
3. I feel that I have a number of good qualities
4. I am able to do things as well as most other people.
5. I feel I do not have much to be proud of. (R)
6. I certainly feel useless at times. (R)
7. I feel that I'm a person of worth, at least on an equal plane with others.
8. I wish I could have more respect for myself (R)
9. All in all, I am inclined to feel that I am a failure (R)
10. I take a positive attitude toward myself.

Response format: 0 = Strongly Disagree, 1 = Disagree, 2 = Agree, 3 = Strongly Agree

Recoded into dichotomies: 0 = Disagree, 1 = Agree

N = 441

Very Brief Introduction to R

Download R for your system

http://cran.stat.ucla.edu

The Comprehensive R Archive Network

Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- Download R for Linux
- Download R for MacOS X
- Download R for Windows

Choose ‘base’

R for Windows

Subdirectories:

base	Binaries for base distribution (managed by Duncan Murdoch). This is what you want to **install R for the first time**.
contrib	Binaries of contributed CRAN packages (for R >= 2.11.x; managed by Uwe Ligges). There is also information on third party software available for CRAN Windows services and corresponding environment and make variables.
old contrib	Binaries of contributed CRAN packages for outdated versions of R (for R < 2.11.x; managed by Uwe Ligges).
Rtools	Tools to build R and R packages (managed by Duncan Murdoch). This is what you want to build your own packages on Windows, or to build R itself.

Please do not submit binaries to CRAN. Package developers might want to contact Duncan Murdoch or Uwe Ligges directly in case of questions / suggestions related to Windows binaries.

You may also want to read the R FAQ and R for Windows FAQ.

Note: CRAN does some checks on these binaries for viruses, but cannot give guarantees. Use the normal precautions with downloaded executables.

Download R

CRAN
Mirrors
What's new?
Task Views
Search

About R
R Homepage
The R Journal

R-3.3.2 for Windows (32/64 bit)

Download R 3.3.2 for Windows (62 megabytes, 32/64 bit)

Installation and other instructions
New features in this version

If you want to double-check that the package you have downloaded matches the package distributed by CRAN, you can compare the md5sum of the .exe to the fingerprint on the master server. You will need a version of md5sum for windows: both graphical and command line versions are available.

Frequently asked questions

- Does R run under my version of Windows?
- How do I update packages in my previous version of R?

Downloading RStudio

https://www.rstudio.com/products/rstudio/download/

- Choose the FREE Open Source Edition

R Studio rstudio::conf Products Resources Pricing About Us Blogs

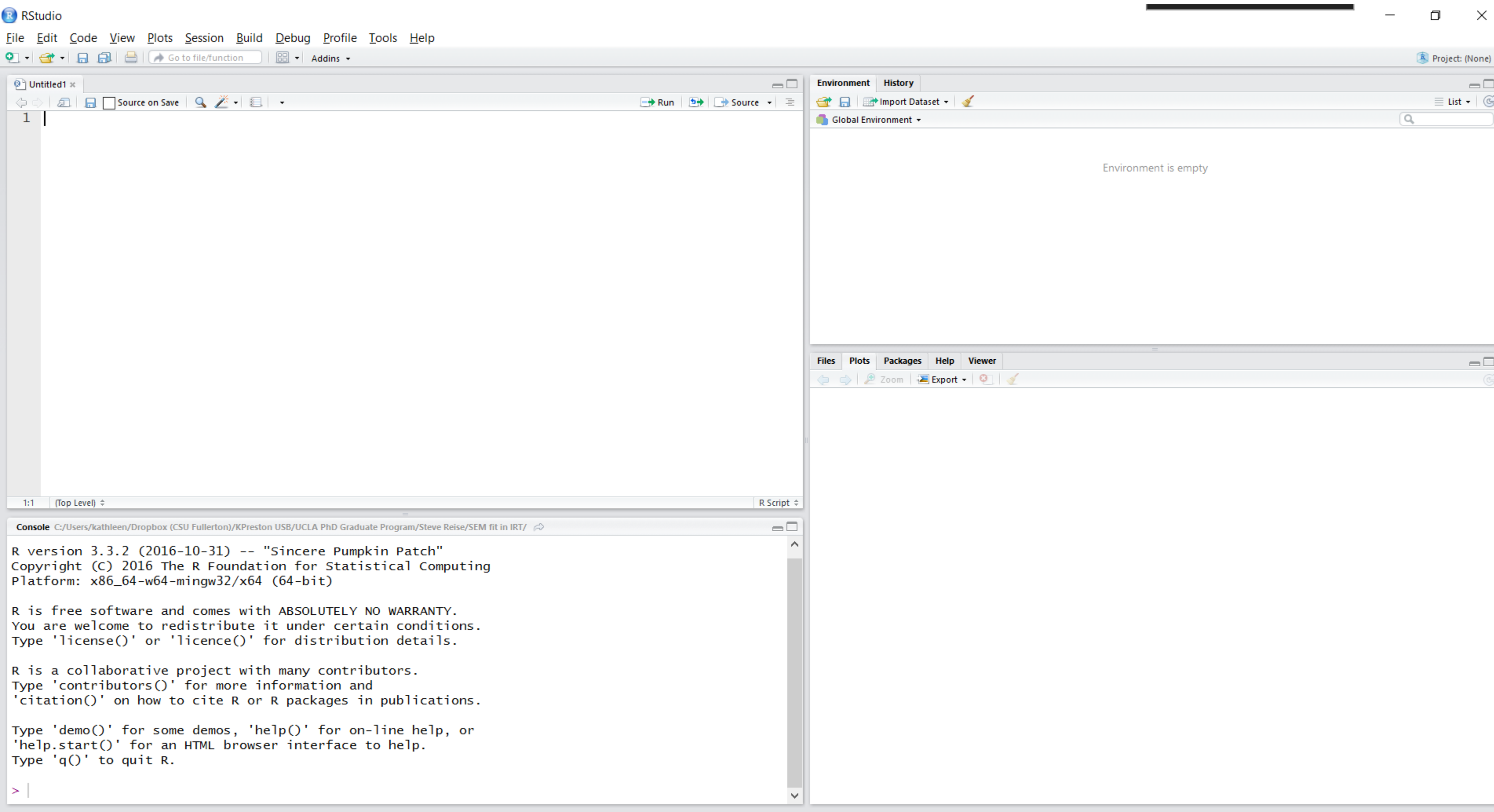
RStudio Desktop 1.0.136 — Release Notes

RStudio requires R 2.11.1+. If you don't already have R, download it here.

Installers for Supported Platforms

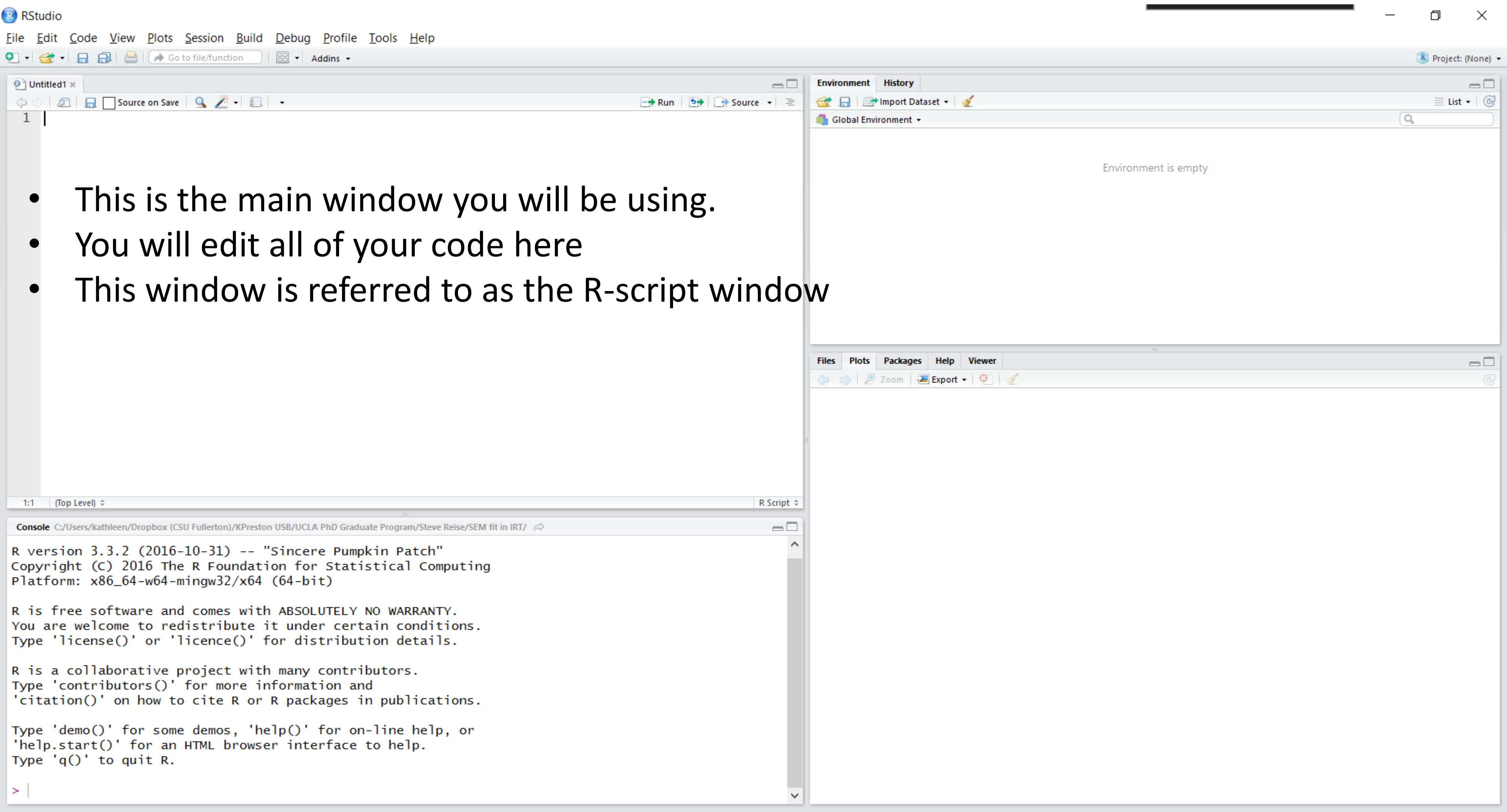
Installers	Size	Date	MD5
RStudio 1.0.136 - Windows Vista/7/8/10	81.9 MB	2016-12-21	93b3f307f567c33f7a4db4c114099b3e
RStudio 1.0.136 - Mac OS X 10.6+ (64-bit)	71.2 MB	2016-12-21	12d6d6ade0203a2fcef6fe3dea65c1ae
RStudio 1.0.136 - Ubuntu 12.04+/Debian 8+ (32-bit)	85.5 MB	2016-12-21	0a20fb89d8aaeb39b329a640ddadd2c5
RStudio 1.0.136 - Ubuntu 12.04+/Debian 8+ (64-bit)	92.1 MB	2016-12-21	2a73b88a12a9fbaf96251cecf8b41340
RStudio 1.0.136 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (32-bit)	84.7 MB	2016-12-21	fa6179a7855bff0f939a34c169da45fd
RStudio 1.0.136 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (64-bit)	85.7 MB	2016-12-21	2b3a148ded380b704e58496befb55545

Navigating RStudio

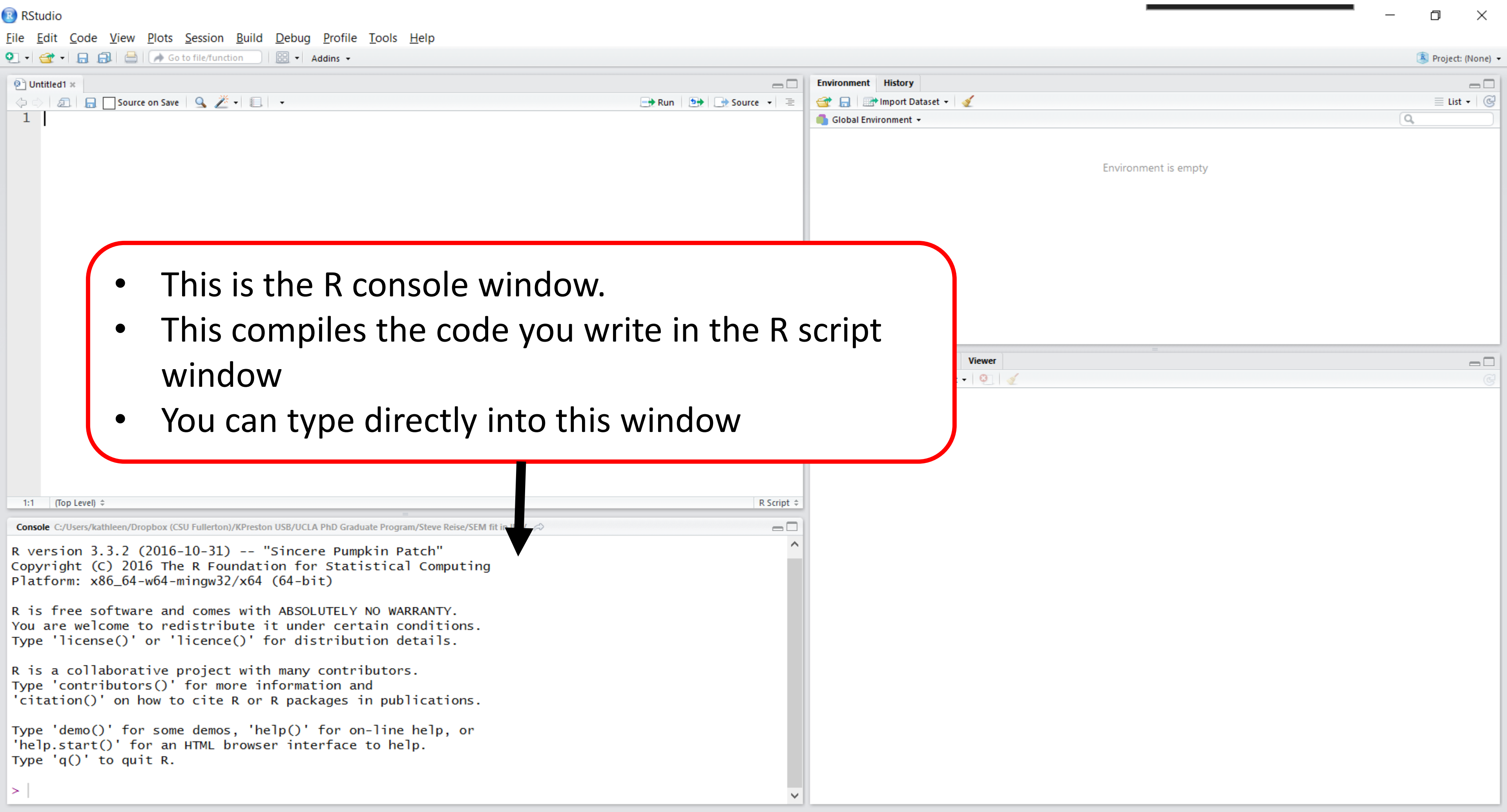


RStudio: R-script

- This is the main window you will be using.
- You will edit all of your code here
- This window is referred to as the R-script window



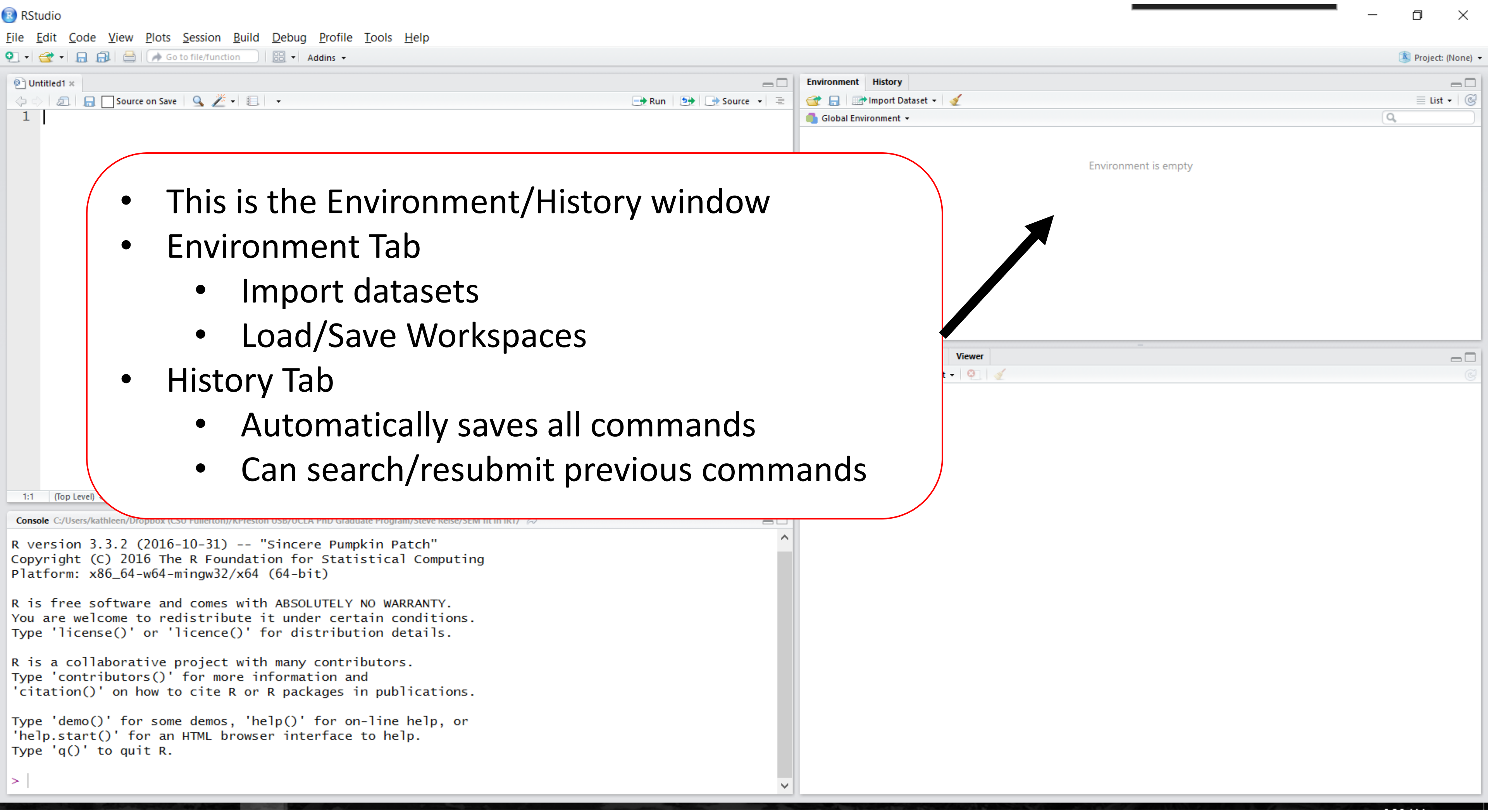
RStudio: R console



- This is the R console window.
- This compiles the code you write in the R script window
- You can type directly into this window

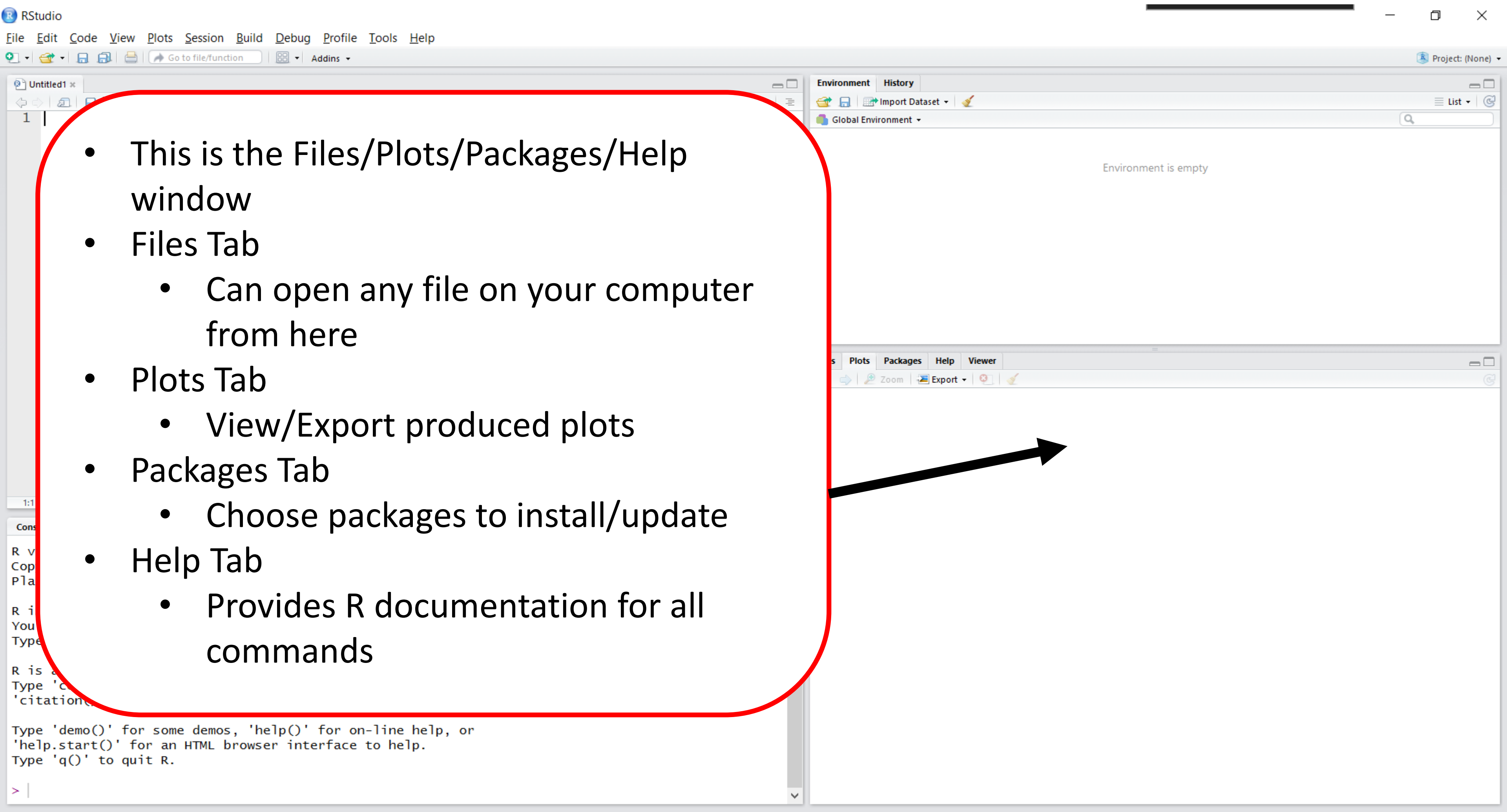
RStudio: Workspace/History

- This is the Environment/History window
- Environment Tab
  - Import datasets
  - Load/Save Workspaces
- History Tab
  - Automatically saves all commands
  - Can search/resubmit previous commands



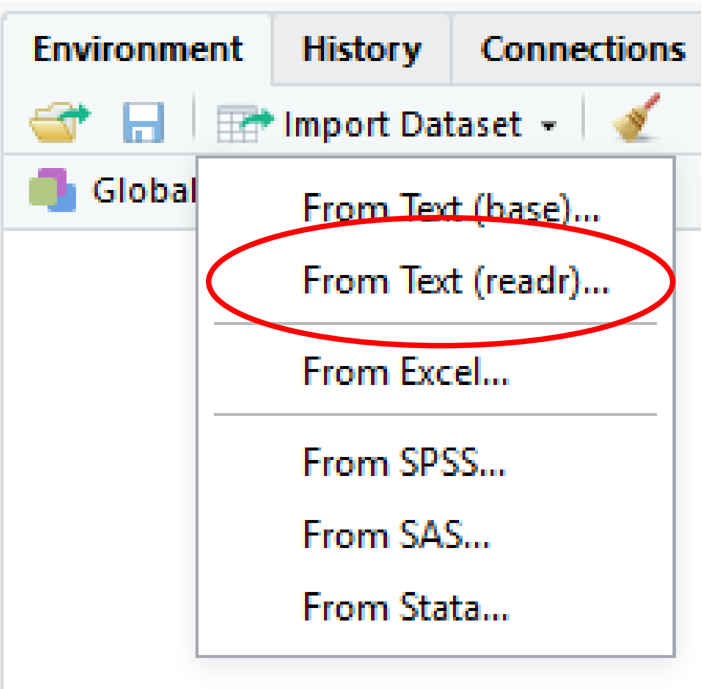
RStudio: Files/Plots/Packages/Help

- This is the Files/Plots/Packages/Help window
- Files Tab
    - Can open any file on your computer from here
- Plots Tab
    - View/Export produced plots
- Packages Tab
    - Choose packages to install/update
- Help Tab
    - Provides R documentation for all commands

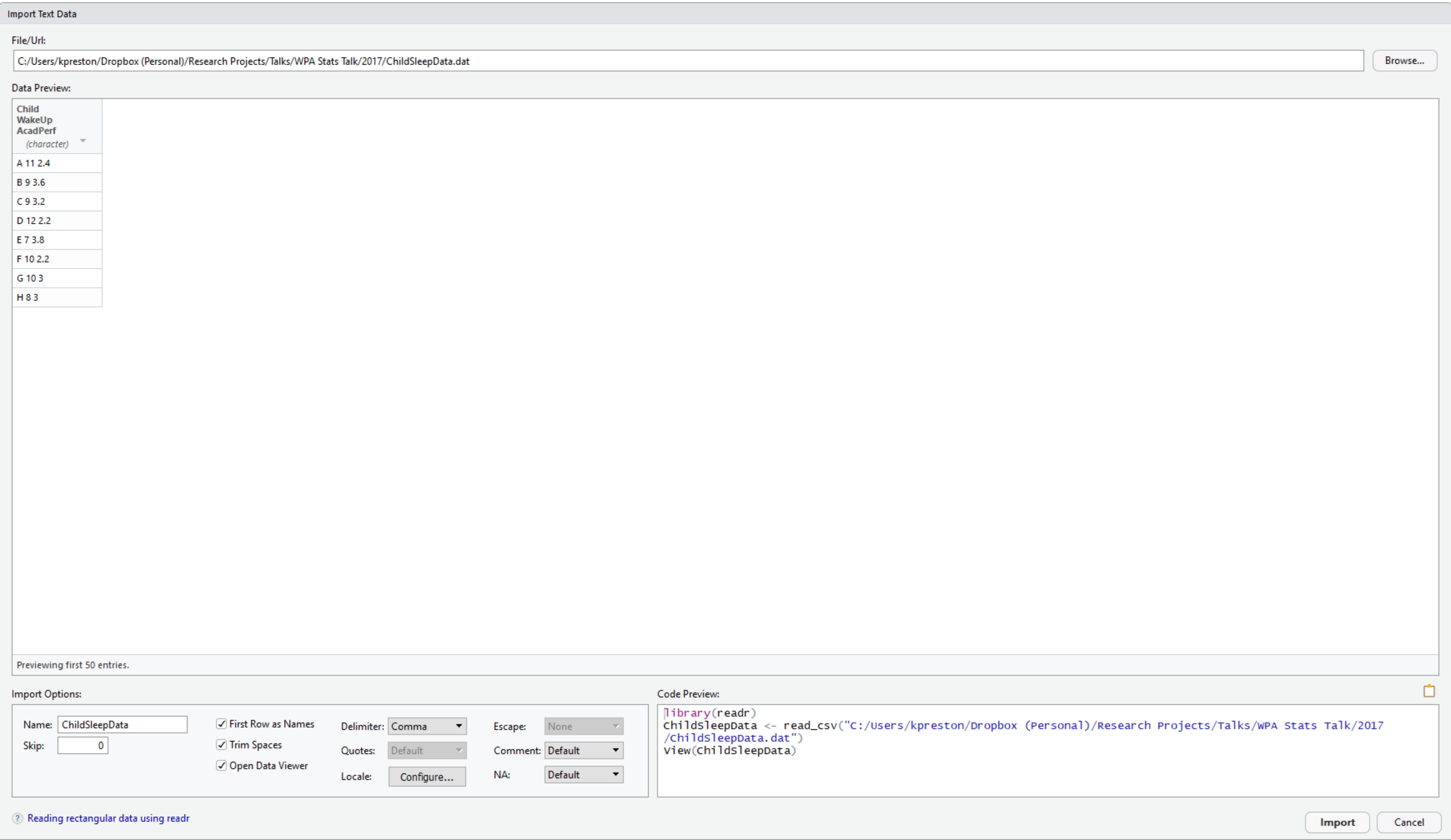


RStudio: Workspace Importing a Dataset

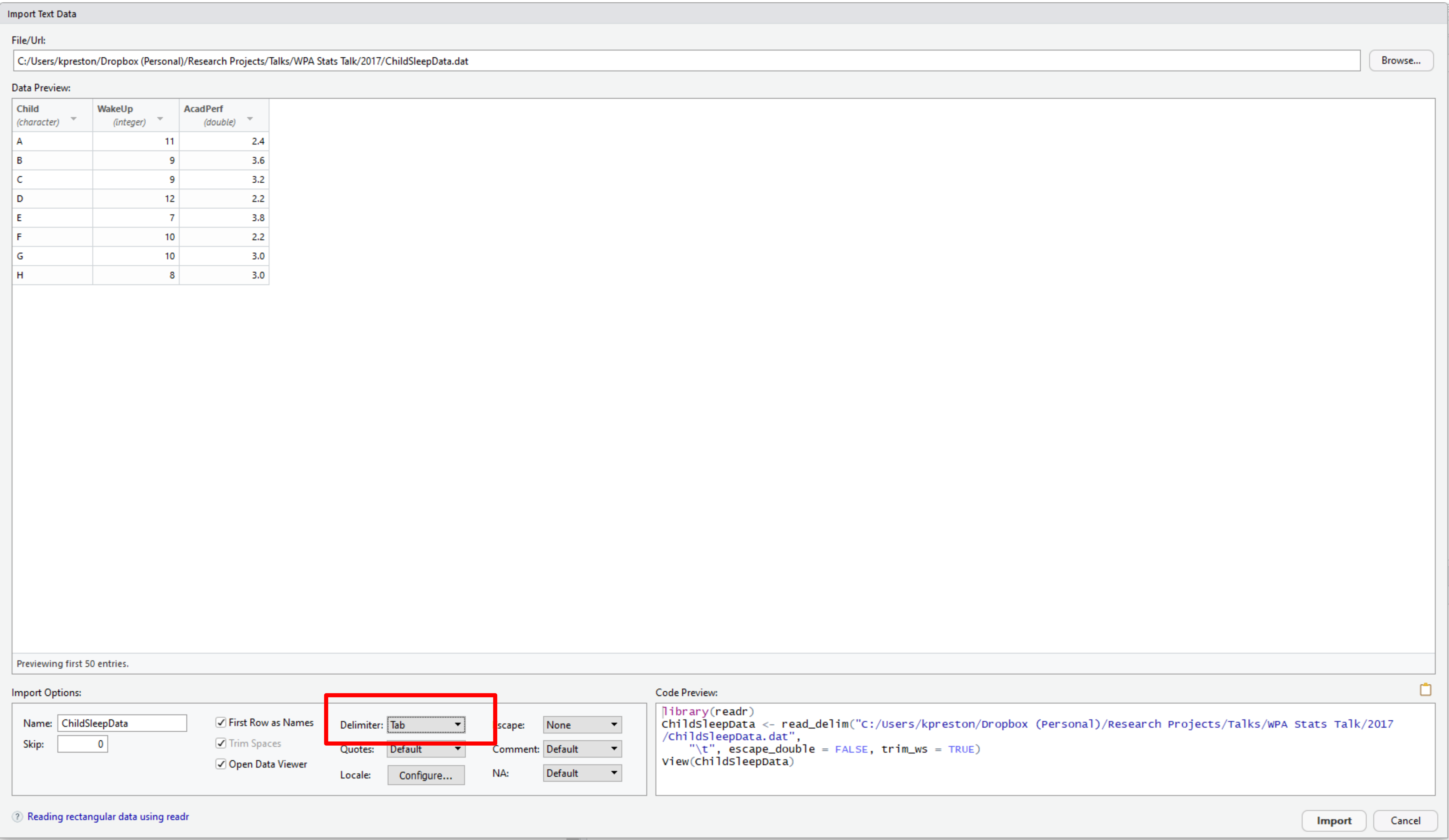
- Import Dataset
  - Can import from the following formats
    - .csv or .dat or .txt
    - .xls or .xlsx
    - .sav
    - .sas7bdat
    - .dta
- Choose ChildSleepData.dat file from computer



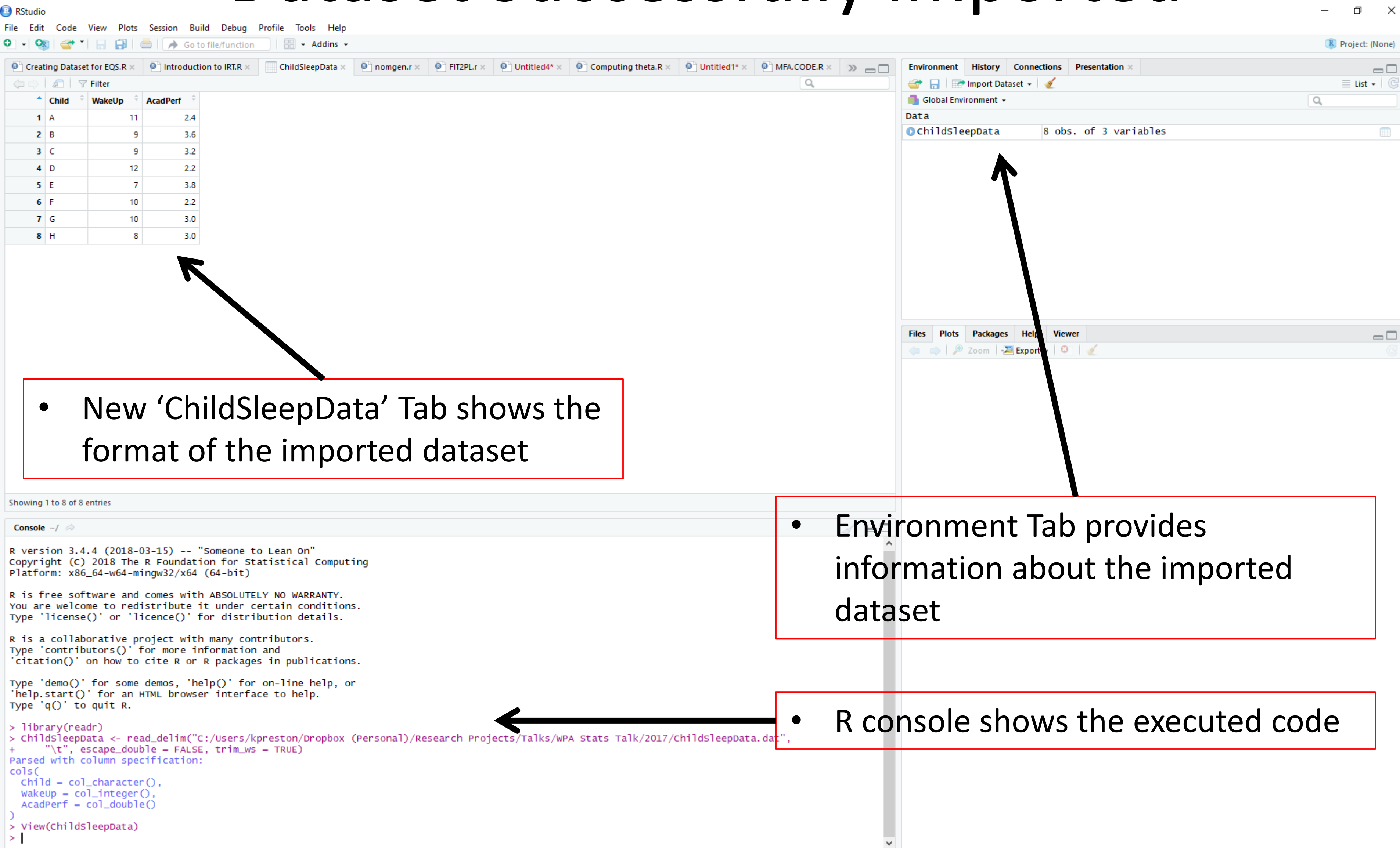
RStudio: Workspace
Importing a Dataset



RStudio: Workspace
Change Delimiter to ‘Tab’



RStudio:
Dataset Successfully Imported

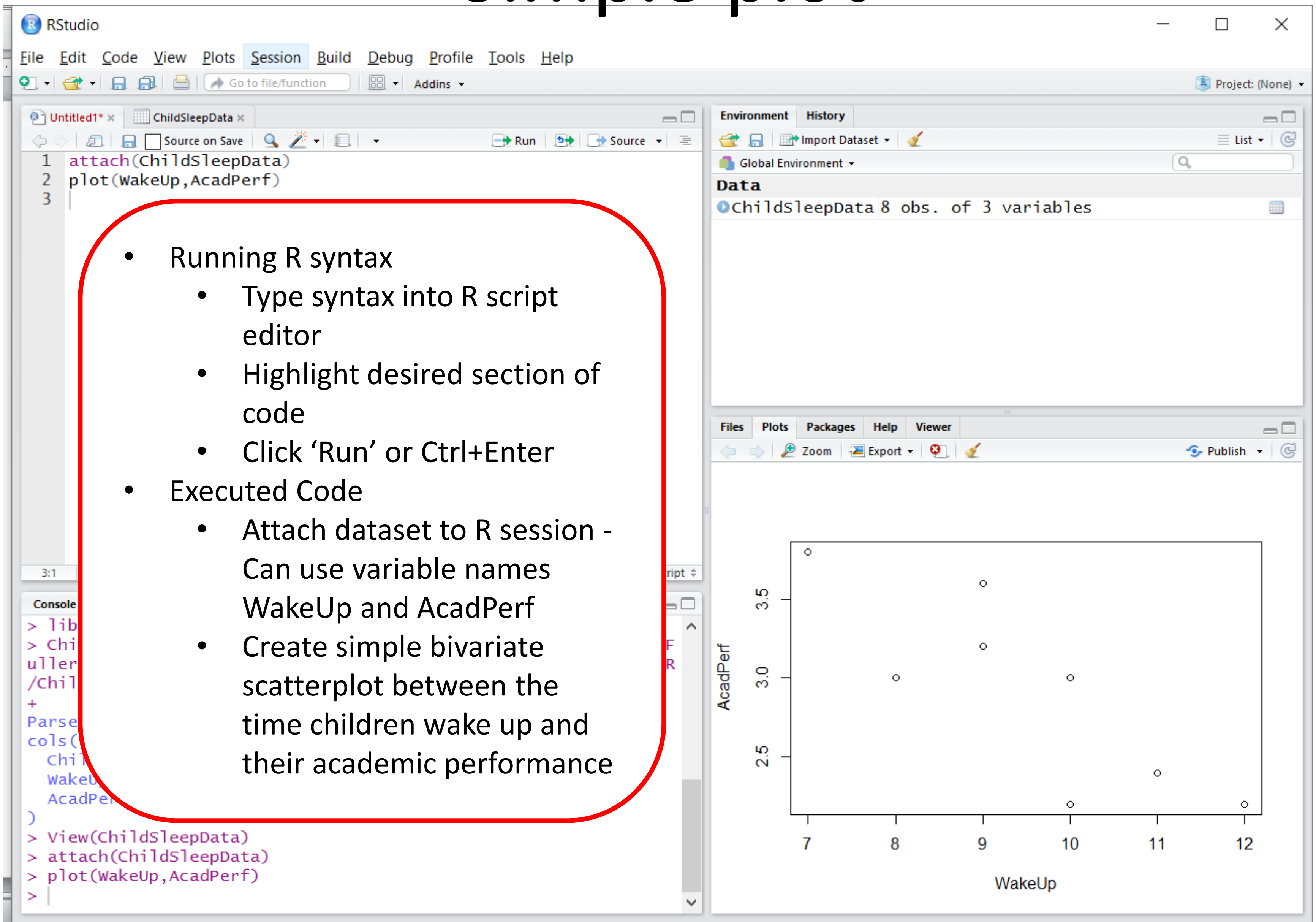


- New ‘ChildSleepData’ Tab shows the format of the imported dataset

- Environment Tab provides information about the imported dataset

- R console shows the executed code

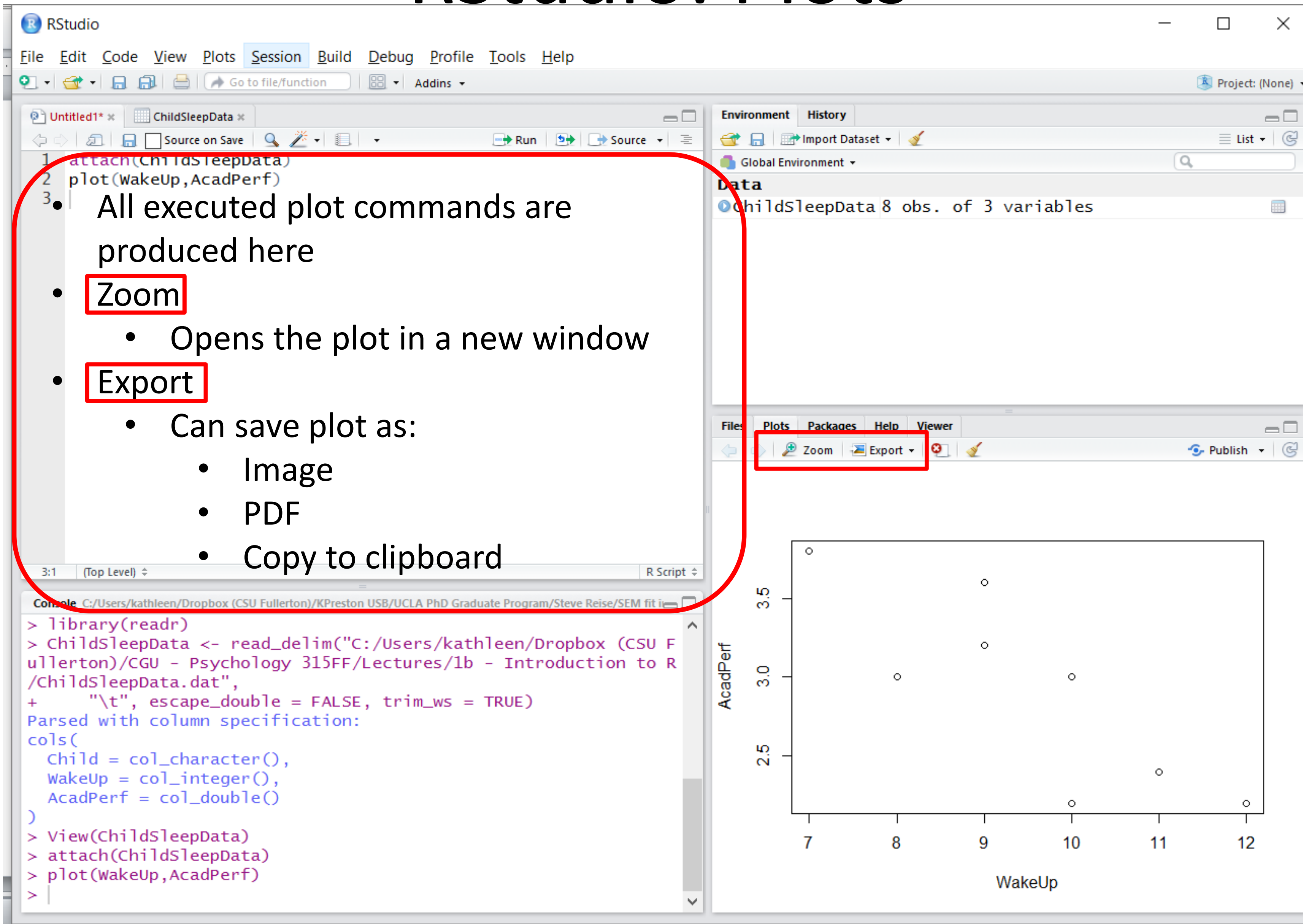
Simple plot



- Running R syntax
  - Type syntax into R script editor
  - Highlight desired section of code
  - Click ‘Run’ or Ctrl+Enter
- Executed Code
  - Attach dataset to R session - Can use variable names WakeUp and AcadPerf
  - Create simple bivariate scatterplot between the time children wake up and their academic performance

RStudio: Plots

- All executed plot commands are produced here
- Zoom
  - Opens the plot in a new window
- Export
  - Can save plot as:
    - Image
    - PDF
    - Copy to clipboard



R Language Basics

Packages

- Base R and most Rpackages are available for download from the Comprehensive R Archive Network (CRAN)
  - cran.r-project.org
  - base R comes with a number of basic data management, analysis, and graphical tools
- We will start by installing a package, car, a package that is frequently used in R tutorials
  - Packages only need to be installed once
  - install.packages(“car”)
- Once the package is installed, it must be loaded
  - Packages must be loaded every time you start an R session
  - library(car)

R Programming Basics

- R is case sensitive.
- The # character at the beginning of a line signifies a comment, which is not executed. Each commented line must start with a #
- Help for function keywords are accessed by the help( ) function with the keyword inside the parentheses
- stores both data and output from data analysis (as well as everything else) in objects
- Things are assigned to and stored in objects using the <- or = operator
- Note: do NOT separate < and - with a space to create <-

DataFrames

```
# Referencing within Data Frames
# single cell value
ChildSleepData[2, 3]
[1] 3.6
# omitting row value implies all rows; here all rows in column 3
ChildSleepData[, 3]
[1] 2.4 3.6 3.2 2.2 3.8 2.2 3.0 3.0
# omitting column values implies all columns; here all columns in row 2
ChildSleepData[2, ]
  Child WakeUp AcadPerf
2     B      9      3.6
# can also use ranges - rows 2 and 3, columns 2 and 3
ChildSleepData[2:3, 2:3]
  WakeUp AcadPerf
2      9      3.6
3      9      3.2
```

Variable Indexing

- We can determine the variable names in the data set with the names function

```
#prints the variable names in the dataset
names(ChildSleepData)
[1] "Child"     "WakeUp"    "AcadPerf"
```

- We can also access variables directly by using their names, either with object[,“variable”] notation or object$variable notation

```
# get first 3 rows of variable female using two methods
ChildSleepData[1:3, "WakeUp"]
[1] 11  9  9
ChildSleepData$WakeUp[1:3]
[1] 11  9  9
```

Exploring the data set

- Using dim( ), we get the number of observations (rows) and variables (columns) in the data set.

```
dim(ChildSleepData)
[1] 8 3
```

- summary( ) is a generic function to summarize many types of R objects, including data sets.
- When used on a data set, summary( ) returns distributional summaries of variables in the data set.

```
summary(ChildSleepData)
```

```
     Child        WakeUp            AcadPerf    
 A      :1   Min.   : 7.00   Min.   :2.200  
 B      :1   1st Qu.: 8.75   1st Qu.:2.350  
 C      :1   Median : 9.50   Median :3.000  
 D      :1   Mean   : 9.50   Mean   :2.925  
 E      :1   3rd Qu.:10.25   3rd Qu.:3.300  
 F      :1   Max.   :12.00   Max.   :3.800  
 (Other):2                                  
```

Preparing R Session

```
install.packages("psych")
install.packages("GPArotation")
install.packages("sirt")
install.packages("ltm")
install.packages("irtoys")
install.packages("mirt")
install.packages("latticeExtra")
```

```
library(psych)
library(GPArotation)
library(sirt)
library(msm)
library(ltm)
library(irtoys)
library(mirt)
library(latticeExtra)
```

Data Entry and Recoding

```
## Read in Raw Data ##
data<- read.delim("C:/Users/kpreston/Self-esteem Dichotomous.dat", header=FALSE)
data[data==9] <-NA #Replace missing code 9 with NA
nitems<-ncol(data)

#recode data
keys <- c(1,-1,1,1,-1,-1,1,-1,-1,1)
data <-
reverse.code(keys,data,mini=rep(0,10),maxi=rep(1,10))
```

Checking Response Frequencies

```
# Check response frequencies

response.frequencies(data)

##              0          1        miss
## V1  0.7573696 0.24263039 0.000000000
## V2- 0.5124717 0.48752834 0.000000000
## V3  0.9183673 0.08163265 0.000000000
## V4  0.9138322 0.08616780 0.000000000
## V5- 0.7977273 0.20227273 0.002267574
## V6- 0.5419501 0.45804989 0.000000000
## V7  0.8820862 0.11791383 0.000000000
## V8- 0.4454545 0.55454545 0.002267574
## V9- 0.8707483 0.12925170 0.000000000
## V10 0.8049887 0.19501134 0.000000000
```

Assumption Checking: Unidimensionality

1. Estimate bifactor model from poly/tetrachoric correlations

```
(rfac <- tetrachoric(data,na.rm=TRUE))
```

```
## Call: tetrachoric(x = data, na.rm = TRUE)
## tetrachoric correlation
##     V1   V2-  V3   V4   V5-  V6-  V7   V8-  V9-  V10
## V1  1.00
## V2- 0.54 1.00
## V3  0.67 0.63 1.00
## V4  0.57 0.46 0.77 1.00
## V5- 0.49 0.60 0.52 0.52 1.00
## V6- 0.44 0.80 0.65 0.38 0.62 1.00
## V7  0.50 0.48 0.76 0.68 0.50 0.46 1.00
## V8- 0.59 0.63 0.56 0.39 0.45 0.60 0.47 1.00
## V9- 0.47 0.58 0.39 0.41 0.62 0.61 0.49 0.65 1.00
## V10 0.81 0.59 0.76 0.62 0.56 0.61 0.70 0.66 0.56 1.00
##
##  with tau of
##     V1    V2-     V3     V4    V5-    V6-     V7    V8-    V9-    V10
##  0.698  0.031  1.394  1.365  0.834  0.105  1.185 -0.137  1.130  0.860
```

Assumption Checking: Unidimensionality

2. Apply a Schmid-Leiman (SL) transformation to the factor loadings to obtain a semi-restricted exploratory hierarchical solution

```
rfac <- tetrachoric(data,na.rm=TRUE)$rho
```

```
(g3 <- schmid(rfac, nfactors = 3, fm = "minres",digits=3,rotate="oblimin"))
```

```
## Schmid-Leiman analysis
## Call: schmid(model = rfac, nfactors = 3, fm = "minres", digits = 3,
##     rotate = "oblimin")
## Schmid Leiman Factor loadings greater than  0.2
##         g   F1*   F2*   F3*   h2   u2   p2
## V1   0.75              0.36 0.71 0.29 0.80
## V2-  0.67  0.52             0.73 0.27 0.62
## V3   0.77        0.52       0.88 0.12 0.68
## V4   0.66        0.53       0.72 0.28 0.61
## V5-  0.61  0.36             0.52 0.48 0.71
## V6-  0.65  0.65             0.84 0.16 0.50
## V7   0.67        0.43       0.64 0.36 0.71
## V8-  0.69  0.28        0.28 0.64 0.36 0.74
## V9-  0.62  0.39             0.58 0.42 0.67
## V10 0.85              0.34 0.87 0.13 0.83
```

Assumption Checking: Unidimensionality

3. ECV$_2$ - Ratio of variance explained by the general factor to variable explained by general and group factors

$$ECV_2 = \frac{\sum(\lambda_G)^2}{\sum(\lambda_G)^2 + \sum(\lambda_{g1})^2 + \sum(\lambda_{g2})^2 + \sum(\lambda_{g3})^2}$$

```
(sumout <- colSums(g3$sl[,1:(ncol(g3$sl)-3)])^2)
##         g      F1*      F2*      F3*
## 48.265261  5.437458  3.444150  1.619778
(ECV2 <- sumout[1] / sum(sumout))
##         g
## 0.8213036
```

- 82.1% of the variance is explained by the general factor

Rasch Model

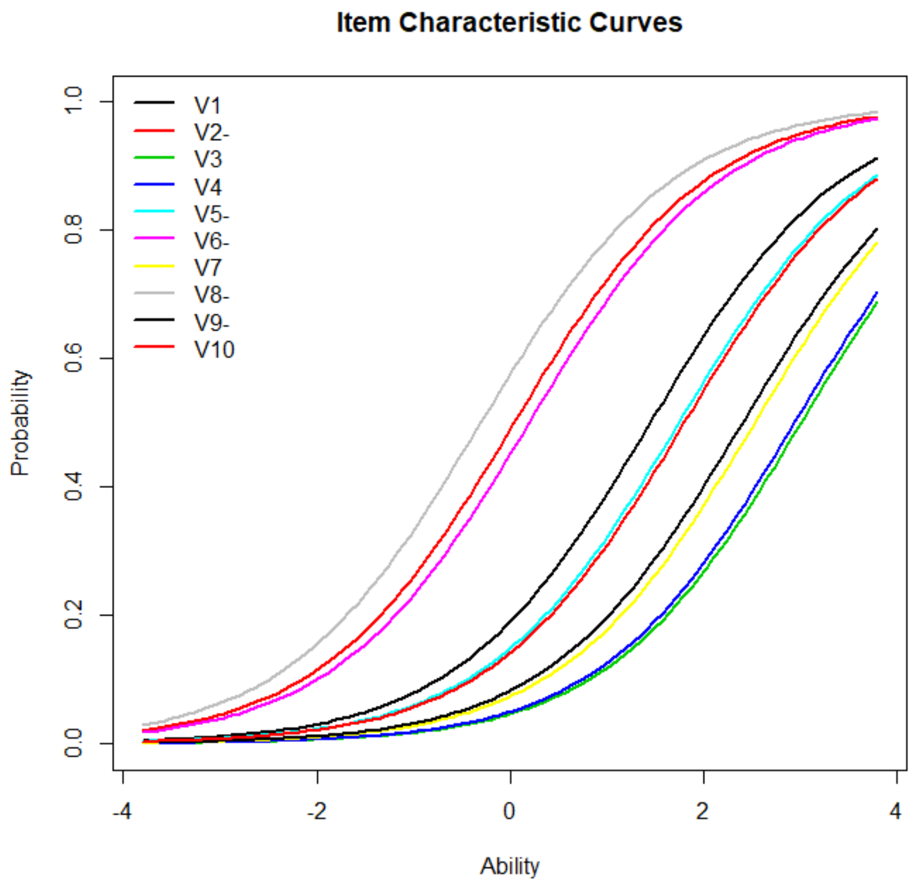
$$P(\theta) = \frac{1}{1 + \exp[-(\theta - b_i)]}$$

Rasch Model

```
modRasch <- rasch(data, IRT.param =
TRUE,constraint=cbind(ncol(data)+1,1))

round(coef(modRasch),2)  # Obtain difficulty and discrimi
nation parameter estimates

##      Dffclt Dscrmn
## V1     1.45      1
## V2-    0.04      1
## V3     3.01      1
## V4     2.94      1
## V5-    1.74      1
## V6-    0.19      1
## V7     2.53      1
## V8-   -0.31      1
## V9-    2.40      1
## V10    1.80      1
```

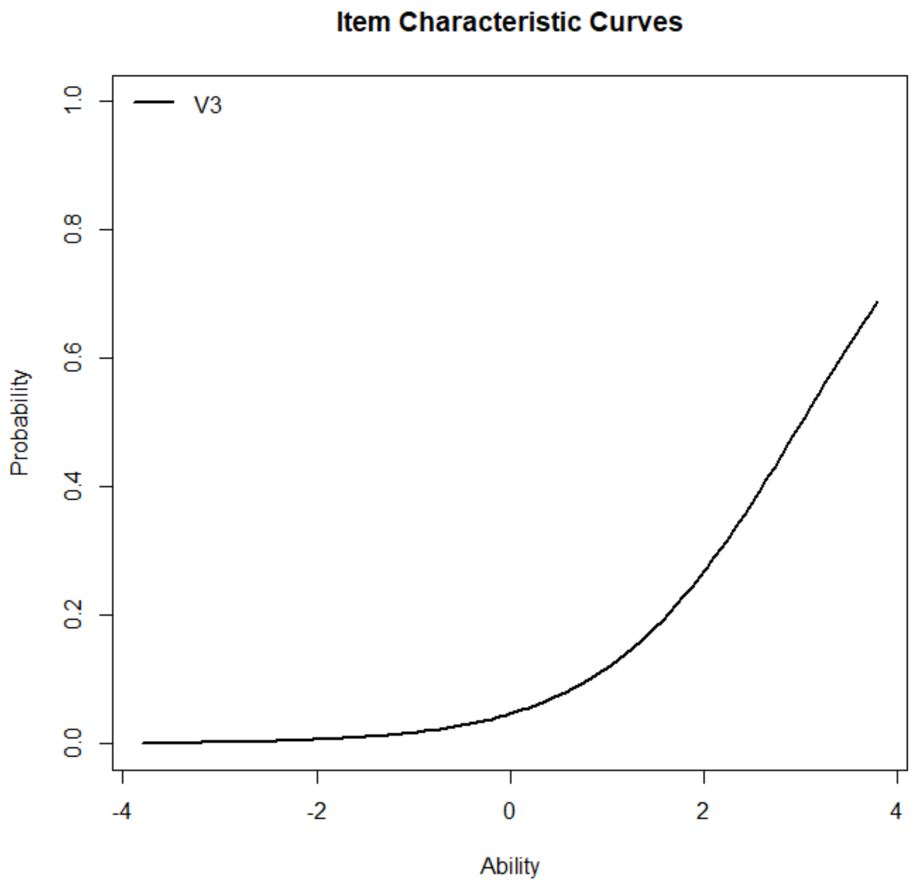
Rasch Model

```
plot(modRasch, type = "ICC", lwd = 2,legend=TRUE)
#Item Characteristic Curves
```



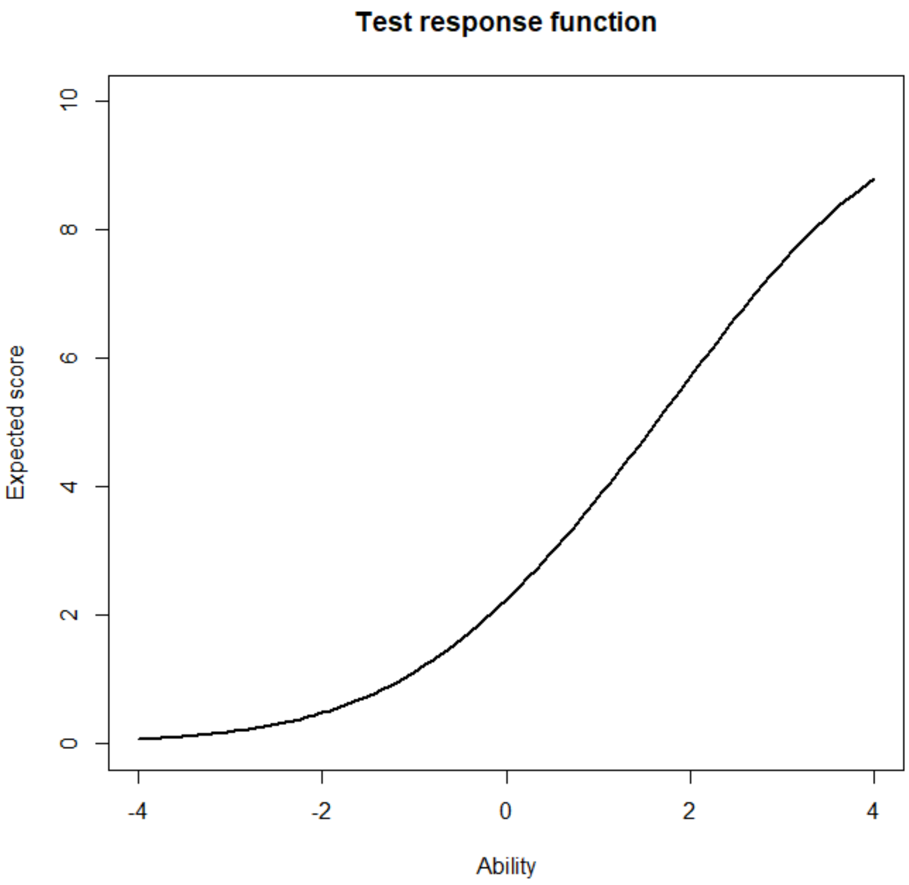
Rasch Model

```
plot(modRasch, type = "ICC", lwd = 2,legend=TRUE,
item=3) #Only Item 3 ICC
```



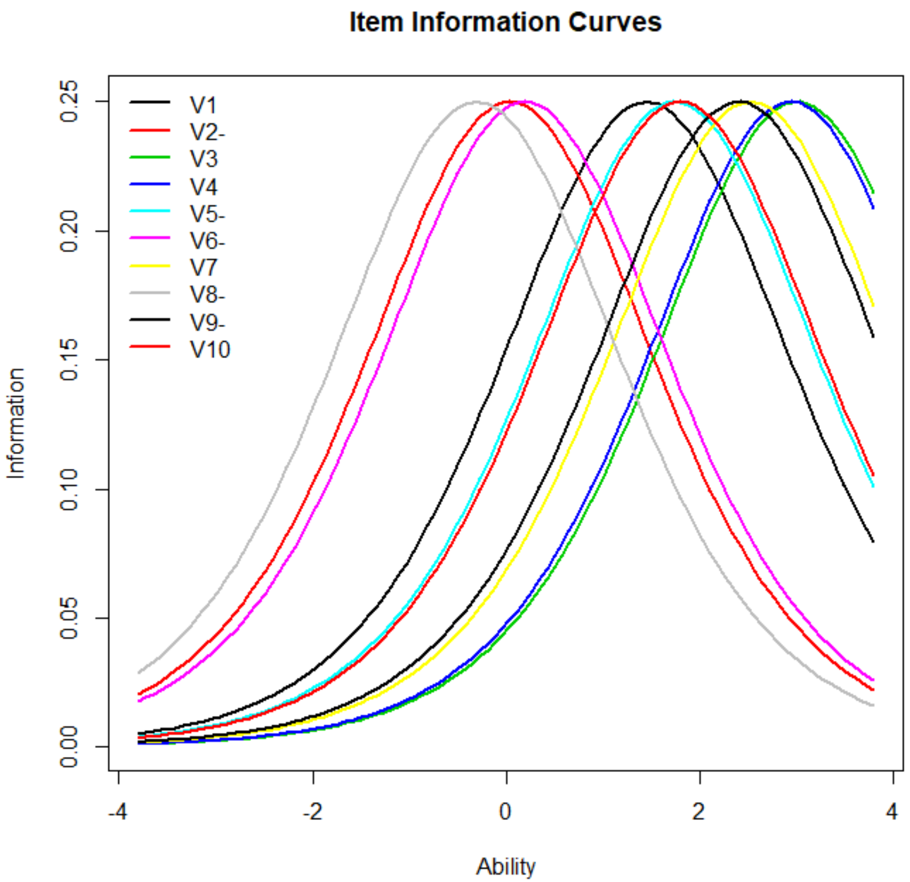
Rasch Model

```
plot(trf(est(data, model = "1PL",rasch="T",engine
= "ltm"))) #Test Response Curve
```



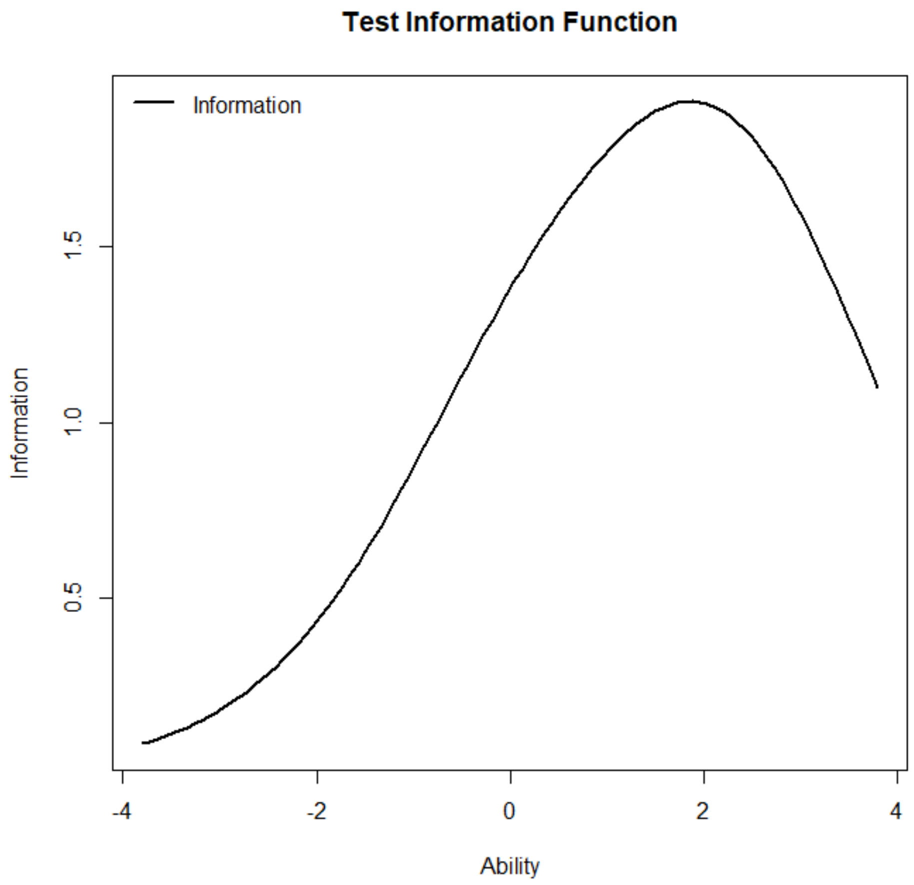
Rasch Model

```
plot(modRasch, type = "IIC", lwd = 2,legend = TRUE) #Item Information Curve
```



Rasch Model

```
plot(modRasch,type = "IIC", lwd = 2,legend =TRUE,
item=0) #Test Information Curve
```



Rasch Model

```
summary(modRasch)

## Model Summary:
##    log.Lik      AIC      BIC
##  -1850.108 3720.216 3761.106
##
## Coefficients:
##              value std.err  z.vals
## Dffclt.V1   1.4478  0.1348 10.7407
## Dffclt.V2-  0.0404  0.1188  0.3402
## Dffclt.V3   3.0063  0.1951 15.4113
## Dffclt.V4   2.9379  0.1910 15.3784
## Dffclt.V5-  1.7446  0.1422 12.2693
## Dffclt.V6-  0.1942  0.1192  1.6287
## Dffclt.V7   2.5267  0.1700 14.8651
## Dffclt.V8- -0.3053  0.1191 -2.5629
## Dffclt.V9-  2.4010  0.1645 14.5949
## Dffclt.V10  1.7999  0.1436 12.5328
## Dscrmn      1.0000      NA      NA
```

Rasch Model

```
thetaRasch <-ability(data,est(data,model="1PL"
,rasch="T",engine="ltm"),method="BME")

round(head(thetaRasch),2)


##          est  sem  n
## [1,]  0.30 0.63 10
## [2,] -0.56 0.69 10
## [3,] -0.56 0.69 10
## [4,]  2.45 0.59 10
## [5,] -1.07 0.74 10
## [6,]  1.06 0.60 10
```

Rasch Model

```
item.fit(modRasch)

## 
## Item-Fit Statistics and P-values
## 
## Alternative: Items do not fit the model
## Ability Categories: 10
## 
##         X^2 Pr(>X^2)
## V1  31.7051  <0.0001
## V2- 68.2809  <0.0001
## V3  40.0526  <0.0001
## V4  23.1139   0.0016
## V5- 31.0865   0.0001
## V6- 59.1941  <0.0001
## V7  27.8279   0.0002
## V8- 27.1124   0.0003
## V9- 24.8152   0.0008
## V10 57.9610  <0.0001
```

1PL

$$P(\theta) = \frac{1}{1 + \exp[-a(\theta - b_i)]}$$

1PL

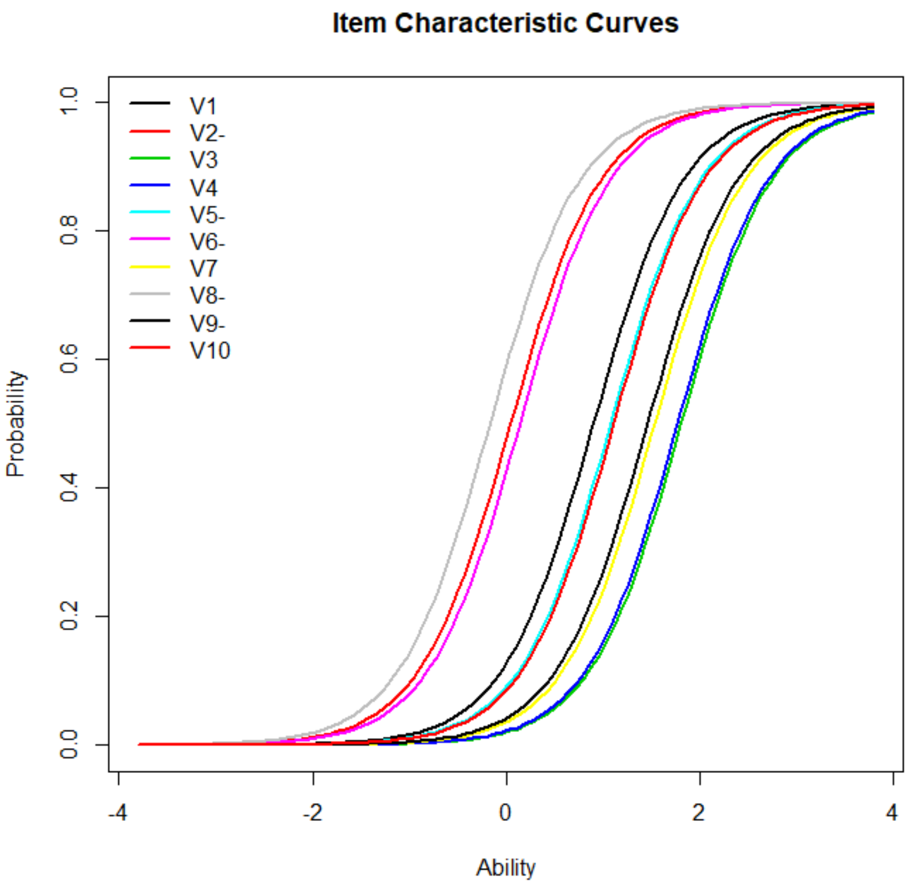
```
mod1PL <- rasch(data, IRT.param = TRUE)

round(coef(mod1PL),2)  # Obtain difficulty and discr
imination parameter estimates

##      Dffclt Dscrmn
## V1     0.90   2.14
## V2-    0.05   2.14
## V3     1.81   2.14
## V4     1.77   2.14
## V5-    1.08   2.14
## V6-    0.14   2.14
## V7     1.53   2.14
## V8-   -0.17   2.14
## V9-    1.46   2.14
## V10    1.11   2.14
```

1PL

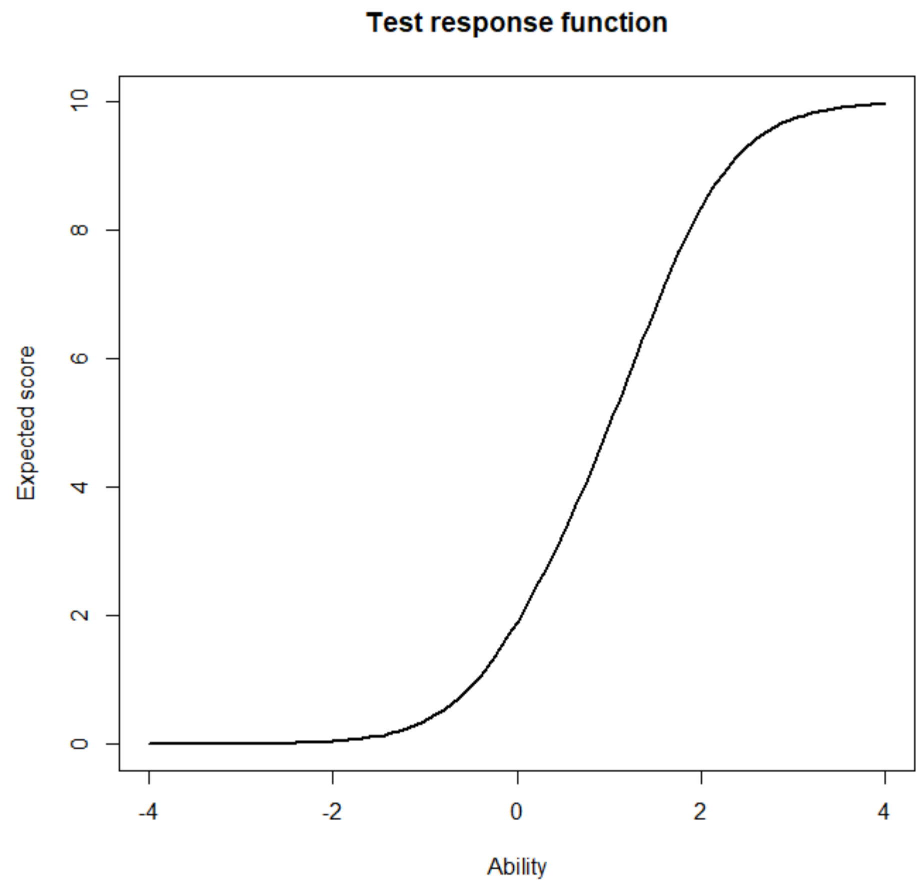
```
summary(mod1PL)
```

```
## Model Summary:
##    log.Lik      AIC      BIC
##  -1762.771 3547.542 3592.522
##
## Coefficients:
##                 value std.err  z.vals
## Dffclt.V1      0.9008  0.0899 10.0225
## Dffclt.V2-     0.0454  0.0772  0.5881
## Dffclt.V3      1.8090  0.1323 13.6714
## Dffclt.V4      1.7700  0.1298 13.6321
## Dffclt.V5-     1.0771  0.0958 11.2414
## Dffclt.V6-     0.1406  0.0773  1.8183
## Dffclt.V7      1.5336  0.1162 13.1996
## Dffclt.V8-    -0.1692  0.0779 -2.1726
## Dffclt.V9-     1.4608  0.1124 12.9933
## Dffclt.V10     1.1093  0.0970 11.4374
## Dscrmn         2.1385  0.1212 17.6424
```

1PL

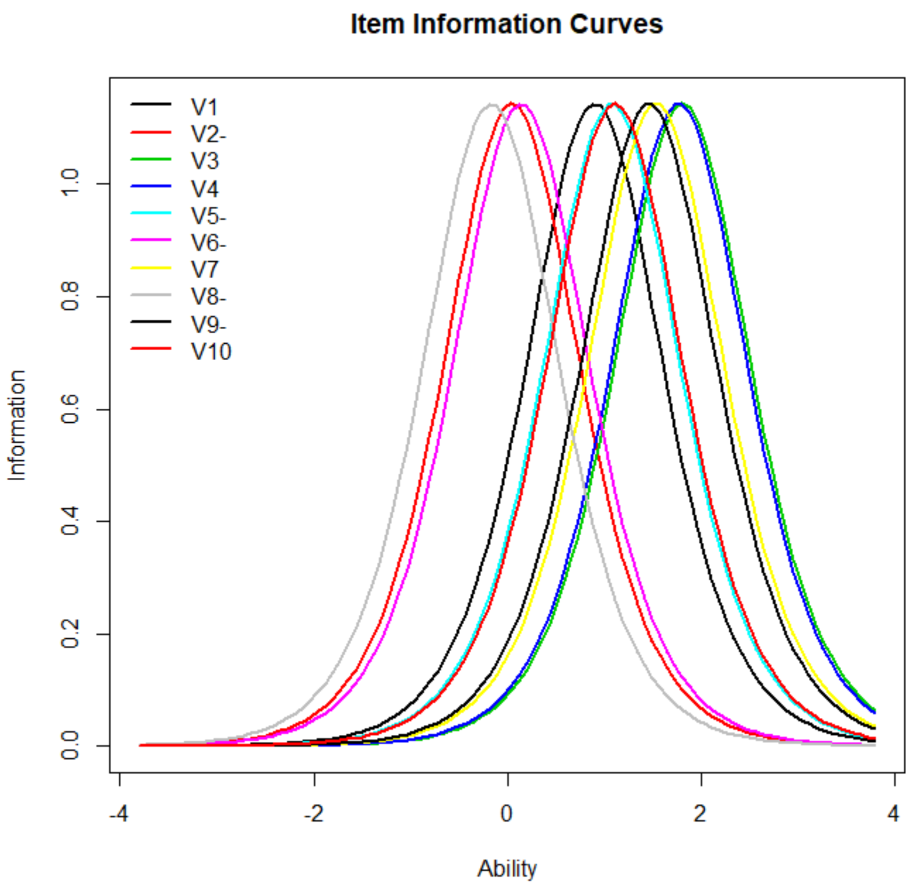


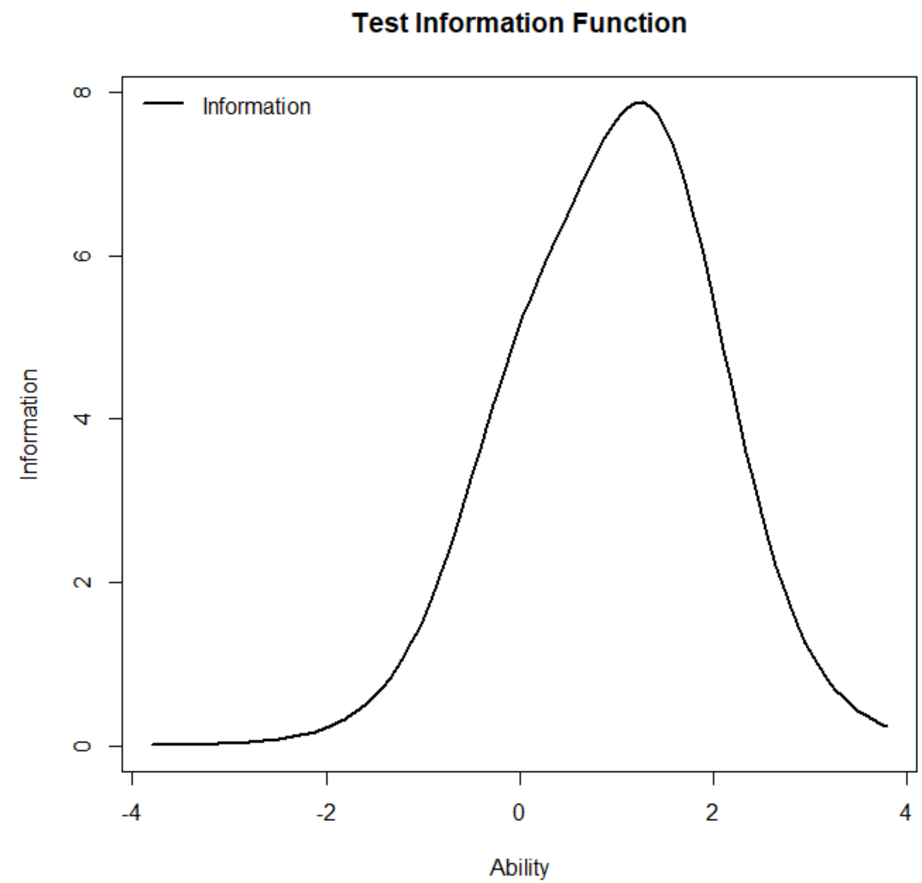
Item Characteristic Curves
V1
V2-
V3
V4
V5-
V6-
V7
V8-
V9-
V10
Probability
0.0
0.2
0.4
0.6
0.8
1.0
-4
-2
0
2
4
Ability



Test response function
Expected score
0
2
4
6
8
10
-4
-2
0
2
4
Ability

1PL


Item Information Curves
V1
V2-
V3
V4
V5-
V6-
V7
V8-
V9-
V10
Information
0.0
0.2
0.4
0.6
0.8
1.0
-4
-2
0
2
4
Ability


Test Information Function
Information
Information
0
2
4
6
8
-4
-2
0
2
4
Ability

1PL

```
item.fit(mod1PL)
```

```
##
## Item-Fit Statistics and P-values
## Alternative: Items do not fit the model
## Ability Categories: 10
##
##         X^2 Pr(>X^2)
## V1   7.5789   0.3712
## V2- 28.5334   0.0002
## V3  18.5744   0.0096
## V4   8.8948   0.2603
## V5- 15.4554   0.0306
## V6- 20.0431   0.0055
## V7  11.1756   0.1311
## V8- 14.8750   0.0376
## V9-  9.2461   0.2355
## V10 23.6501   0.0013
```

Comparing Models

```
## Compare Model fit
anova.rasch(modRasch,mod1PL)

##
##  Likelihood Ratio Table
##                AIC     BIC  log.Lik    LRT df p.value
## modRasch 3720.22 3761.11 -1850.11
## mod1PL   3547.54 3592.52 -1762.77 174.67  1  <0.001
```

2PL

$$P(\theta) = \frac{1}{1 + \exp[-a_i(\theta - b_i)]}$$

2PL

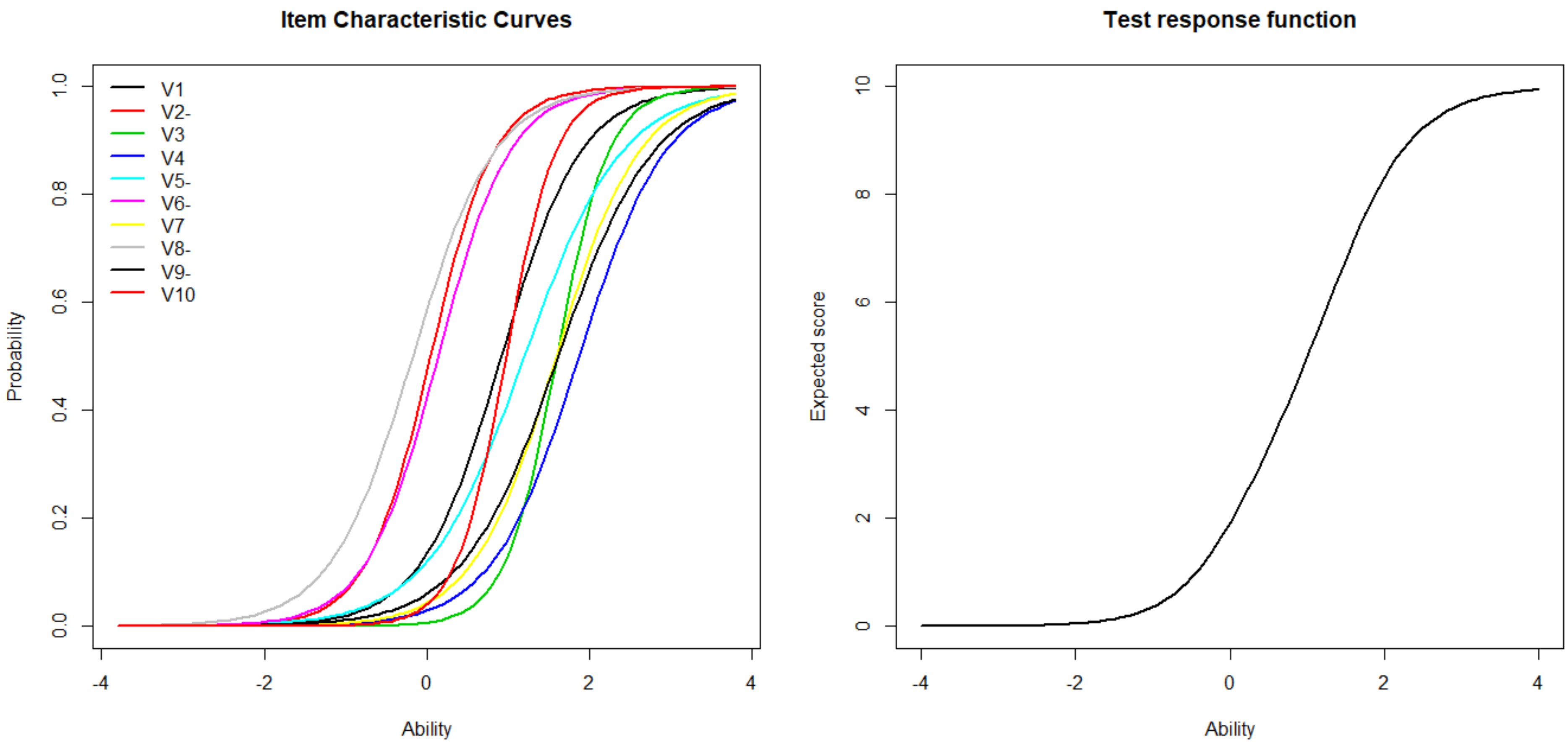
```
# Perform the 2PL analysis with ltm(), and save the
results in "mod2PL“
mod2PL <- ltm(data ~ z1, IRT.param = TRUE)

round(coef(mod2PL),2)  # Obtain difficulty and discr
imination parameter estimates

##      Dffclt Dscrmn
## V1     0.92   2.04
## V2-    0.05   2.55
## V3     1.60   3.12
## V4     1.87   1.87
## V5-    1.20   1.66
## V6-    0.14   2.27
## V7     1.59   1.95
## V8-   -0.17   1.98
## V9-    1.62   1.70
## V10    0.97   3.26
```

2PL

```r
summary(mod2PL)
##
## Model Summary:
##    log.Lik      AIC    BIC
##  -1752.659 3545.319 3627.1
##
## Coefficients:
##               value std.err  z.vals
## Dffclt.V1    0.9166  0.1021  8.9815
## Dffclt.V2-   0.0457  0.0727  0.6281
## Dffclt.V3    1.6011  0.1340 11.9451
## Dffclt.V4    1.8733  0.1998  9.3748
## Dffclt.V5-   1.2008  0.1346  8.9246
## Dffclt.V6-   0.1392  0.0758  1.8353
## Dffclt.V7    1.5907  0.1593  9.9862
## Dffclt.V8-  -0.1740  0.0807 -2.1563
## Dffclt.V9-   1.6169  0.1752  9.2270
## Dffclt.V10   0.9742  0.0896 10.8790
```

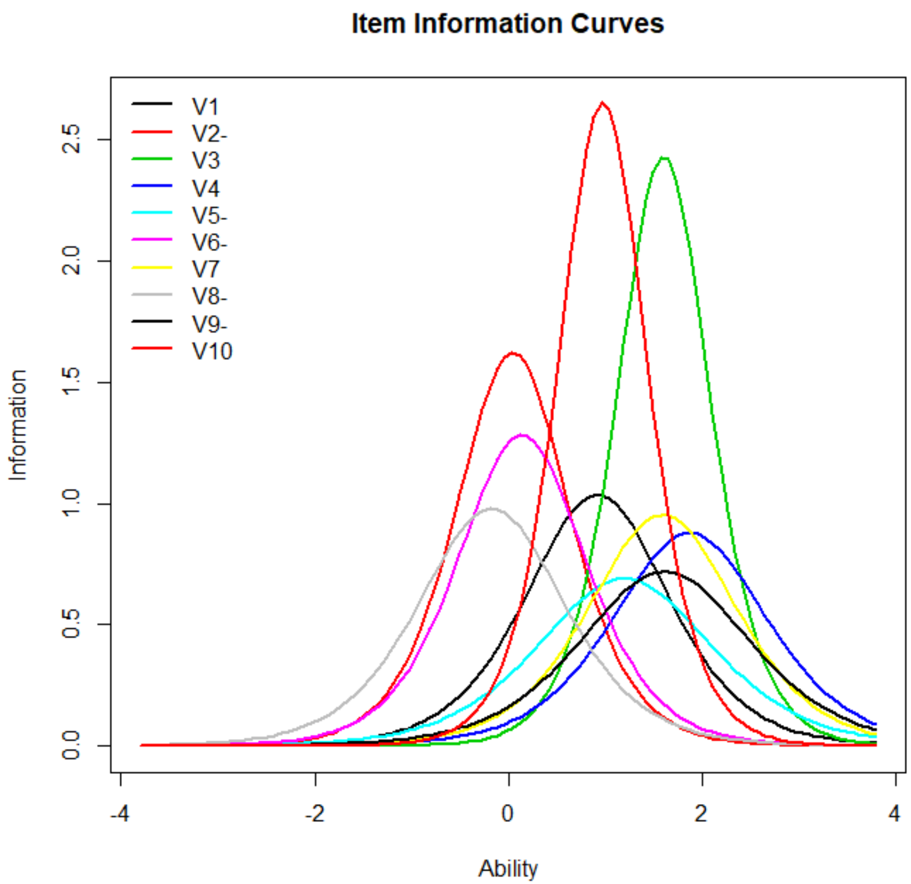
```r
## Dscrmn.V1    2.0359  0.2949  6.9049
## Dscrmn.V2-   2.5474  0.3565  7.1459
## Dscrmn.V3    3.1185  0.6073  5.1354
## Dscrmn.V4    1.8743  0.3265  5.7402
## Dscrmn.V5-   1.6615  0.2396  6.9347
## Dscrmn.V6-   2.2654  0.3124  7.2518
## Dscrmn.V7    1.9536  0.3105  6.2920
## Dscrmn.V8-   1.9781  0.2641  7.4904
## Dscrmn.V9-   1.6982  0.2697  6.2960
## Dscrmn.V10   3.2559  0.5954  5.4685
```

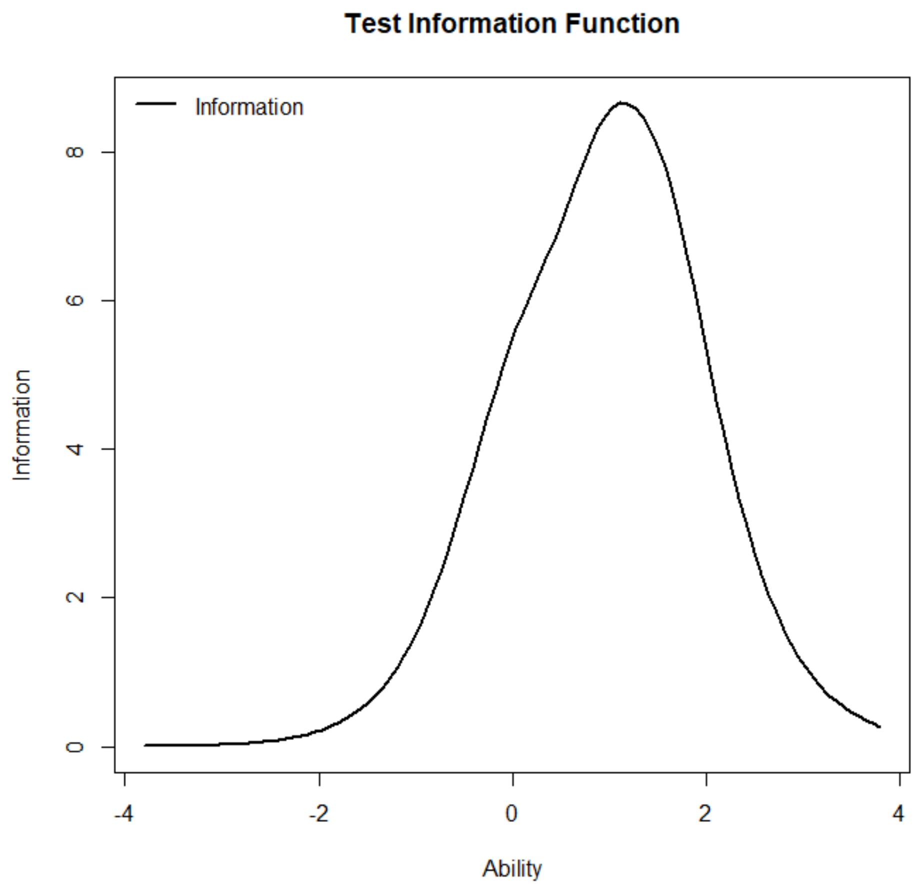
2PL



Item Characteristic Curves
Probability
Ability
V1
V2-
V3
V4
V5-
V6-
V7
V8-
V9-
V10
0.0
0.2
0.4
0.6
0.8
1.0
-4
-2
0
2
4
Test response function
Expected score
Ability
0
2
4
6
8
10
-4
-2
0
2
4

2PL


Item Information Curves
V1
V2-
V3
V4
V5-
V6-
V7
V8-
V9-
V10
Information
0.0
0.5
1.0
1.5
2.0
2.5
-4
-2
0
2
4
Ability


Test Information Function
Information
Information
0
2
4
6
8
-4
-2
0
2
4
Ability

2PL

```
item.fit(mod2PL)
```

```
##
## Item-Fit Statistics and P-values
##
## Alternative: Items do not fit the model
## Ability Categories: 10
##
##          X^2 Pr(>X^2)
## V1  14.7882   0.0634
## V2- 32.8034   0.0001
## V3  15.1024   0.0572
## V4  19.3237   0.0132
## V5- 22.9620   0.0034
## V6-  8.7313   0.3655
## V7  13.1027   0.1084
## V8- 12.4854   0.1308
## V9- 29.2589   0.0003
## V10 15.4248   0.0514
```

Comparing Models

```
# Compare Model fit
anova.rasch(mod1PL,mod2PL)

##
##  Likelihood Ratio Table
##              AIC     BIC  log.Lik   LRT df p.value
## mod1PL 3547.54 3592.52 -1762.77
## mod2PL 3545.32 3627.10 -1752.66 20.22  9   0.017
```

Scoring
BME (Bayesian Modal Estimation)

Person 1: 1, 1, 0, 1, 0, 0, 0, 0, 0, 0

```
person1 <- data[1,]
## Computing theta
theta <- seq(-4,4,.001)
apar <- coef(mod2PL)[,2]
bpar <- coef(mod2PL)[,1]

P <- matrix(0,nitems,length(theta))
L <- matrix(0,nitems,length(theta))
dis <- dnorm(theta)
dis <- dis/sum(dis)

for(m in 1:nitems){
  P[m,] <- 1/(1+exp(-(apar[m]*(theta-bpar[m]))))}
```

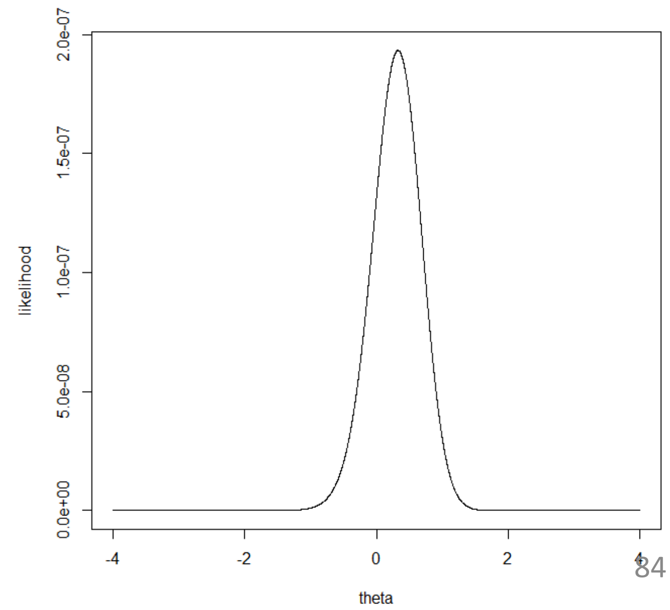
$$L = \prod_{i=1}^{nitems} P_i\,(u_i|\theta)\phi(\theta)$$

$$P(\theta) = \frac{1}{1 + \exp[-a_i(\theta - b_i)]}$$

Scoring
BME (Bayesian Modal Estimation)

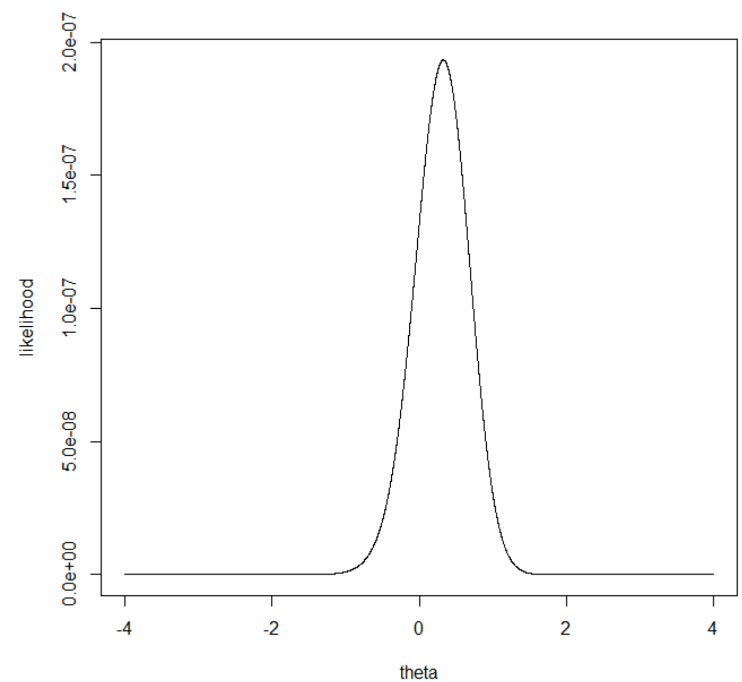
$$L = \prod_{i=1}^{nitems} P_i\,(u_i|\theta)\phi(\theta)$$

```
for(k in 1:length(person1)){
  if(person1[k] == 1){ L[k,] <- P[k,] }
  else if(person1[k] == 0) {L[k,] <- 1-P[k,] }}

likelihood <- dis
for(j in 1:length(person1)){
  likelihood <- likelihood*L[j,]}

plot(theta,likelihood,type="l")

(bme<-theta[which.max(likelihood)])

## [1] 0.325
```



Scoring
BME (Bayesian Modal Estimation)

```
for(k in 1:length(person1)){
  if(person1[k] == 1){ L[k,] <- P[k,] }
  else if(person1[k] == 0) {L[k,] <- 1-P[k,] }}


likelihood <- dis
for(j in 1:length(person1)){
  likelihood <- likelihood*L[j,]}


plot(theta,likelihood,type="l")

(bme<-theta[which.max(likelihood)])

## [1] 0.325

round(theta2PL[1,],2)

##    est   sem     n
##   0.32  0.36 10.00
```

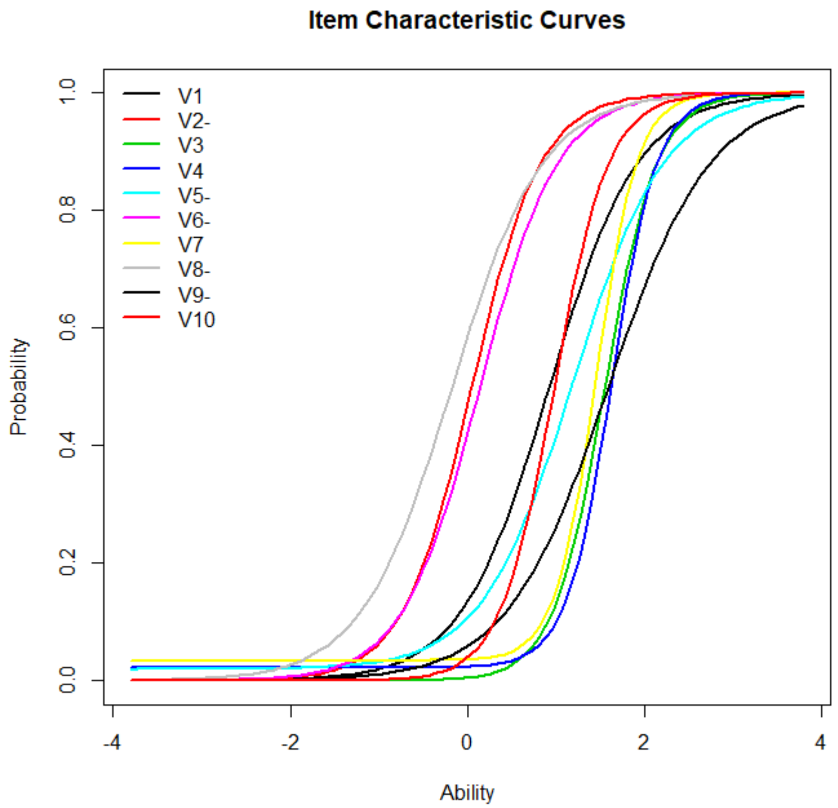


3PL

$$P(\theta) = c_i + \frac{1 - c_i}{1 + \exp[-a_i(\theta - b_i)]}$$

3PL

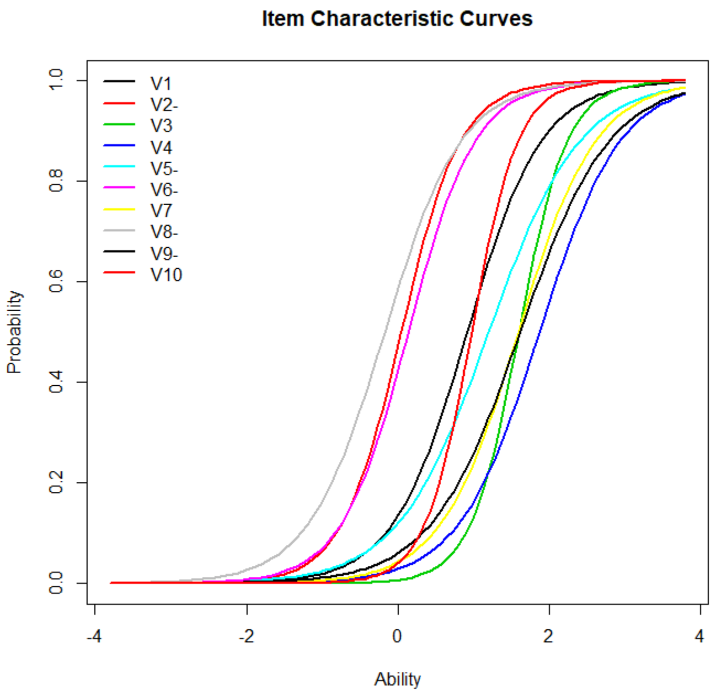
```
mod3PL <- tpm(data, type="latent.trait",IRT.param = TRUE,
control=c(optimizer="nlminb"))

round(coef(mod3PL),2)  # Obtain difficulty and discrimina
tion parameter estimates

##       Gussng Dffclt Dscrmn
## V1      0.00   0.92   2.01
## V2-     0.00   0.05   2.56
## V3      0.00   1.55   3.38
## V4      0.02   1.63   3.88
## V5-     0.02   1.20   1.94
## V6-     0.00   0.14   2.29
## V7      0.03   1.46   4.30
## V8-     0.00  -0.17   1.95
## V9-     0.00   1.59   1.74
## V10     0.00   0.98   3.25
```

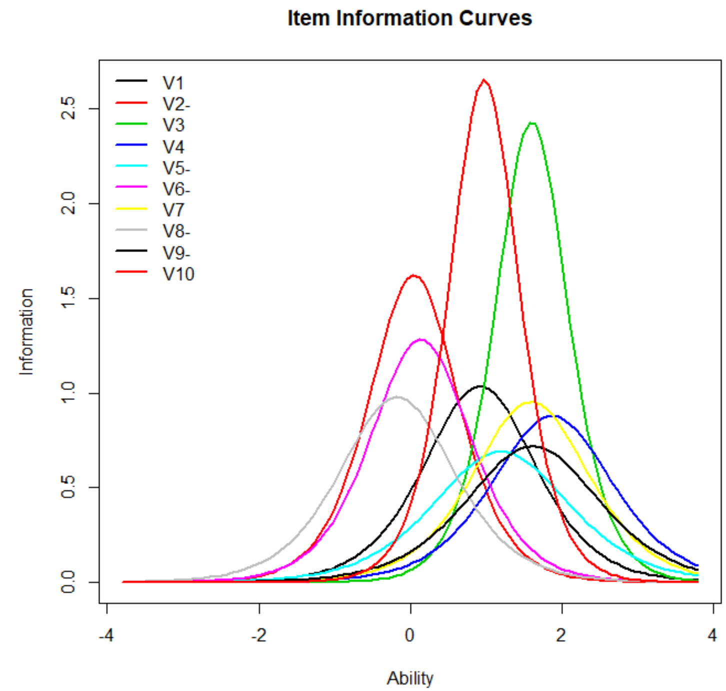


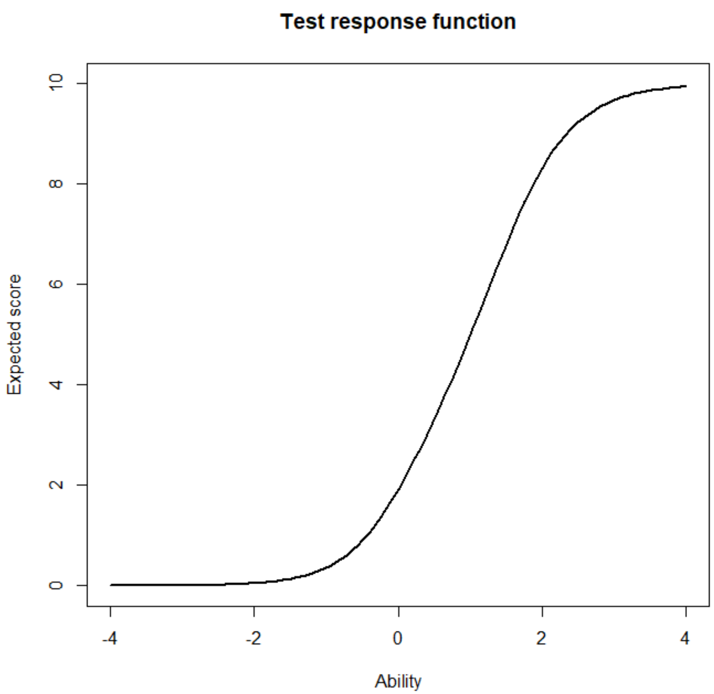
Best Model

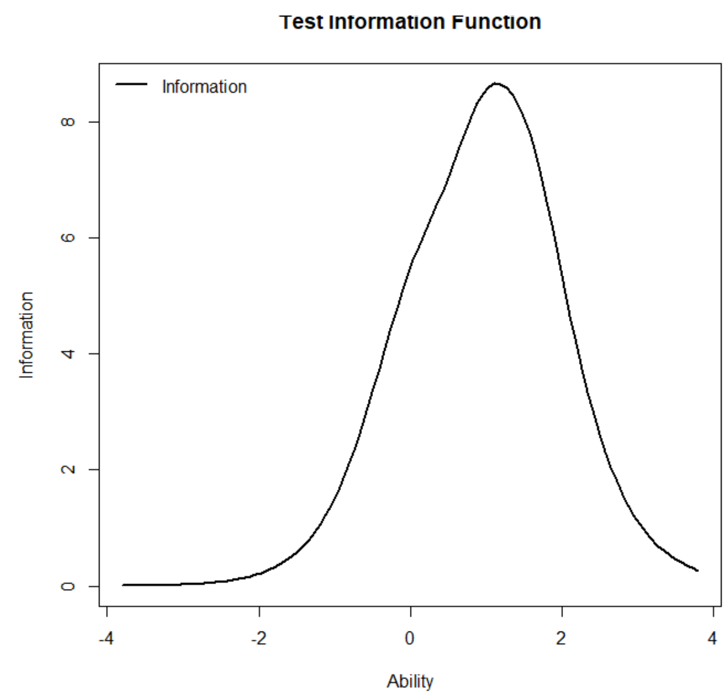
2PL

2PL


Item Characteristic Curves
Probability
Ability
V1
V2-
V3
V4
V5-
V6-
V7
V8-
V9-
V10


Item Information Curves
Information
Ability
V1
V2-
V3
V4
V5-
V6-
V7
V8-
V9-
V10


Test response function
Expected score
Ability


Test Information Function
Information
Ability

Confirm Unidimensionality: Modified Parallel Analysis

```
unidimTest(mod2PL)
```

```
##
## Unidimensionality Check using Modified Parallel Analysis
##
##
## Alternative hypothesis: the second eigenvalue of the observed
## data is substantially larger than the second eigenvalue of
## data under the assumed IRT model
##
## Second eigenvalue in the observed data: 0.8961
## Average of second eigenvalues in Monte Carlo samples: 0.6274
## Monte Carlo samples: 100
## p-value: 0.0594
```

Assessing Local Independence

```
## Assessing Local Dependence using Yen's Q3 statistic, (Yen, 1984)
mod.q3 <- Q3( dat=data, theta=theta2PL[,1] , b=coef(mod2PL)[,1])


## Yen's Q3 Statistic based on an estimated theta score
## *** 10 Items | 45 item pairs
## *** Q3 Descriptives
##       M     SD    Min    10%    25%    50%    75%    90%    Max
## -0.014  0.114 -0.186 -0.131 -0.091 -0.029  0.031  0.138  0.301
```

Yen's Q3 Correlations

```
mod.q3$q3.matrix[upper.tri(mod.q3$q3.matrix,diag=TRUE)] <-0
round(mod.q3$q3.matrix,2)
```

```
##         V1   V2-    V3    V4   V5-   V6-    V7   V8-   V9- V10
## V1
## V2- -0.06
## V3   0.01 -0.15
## V4   0.01 -0.13  0.22
## V5- -0.03  0.02 -0.09  0.00
## V6- -0.15  0.30 -0.11 -0.17  0.06
## V7  -0.07 -0.12  0.18  0.15 -0.03 -0.13
## V8-  0.01  0.06 -0.13 -0.13 -0.08  0.04 -0.09
## V9- -0.06 -0.03 -0.19 -0.08  0.12  0.01 -0.03  0.03
## V10  0.29 -0.11  0.06  0.00 -0.03 -0.07  0.10 -0.01 -0.03
```

```
#Criteria relative to average observed residual correlation from Christensen
, Maransky & Horton (2017)
mod.q3$Q3.stat
```

```
##           M          SD         Min         10%         25%         50%
## -0.01400869  0.11426811 -0.18632406 -0.13112532 -0.09130349 -0.02858531
##         75%         90%         Max
##  0.03106187  0.13792324  0.30136573
```

```
mod.q3$Q3.stat[9]-abs(mod.q3$Q3.stat[1])
## 0.287357
```

Rosenberg Self-Esteem Scale (1965)

1. On the whole, I am satisfied with myself.
2. At times, I think I am not good at all (R)
3. I feel that I have a number of good qualities
4. I am able to do things as well as most other people.
5. I feel I do not have much to be proud of. (R)
6. I certainly feel useless at times. (R)
7. I feel that I'm a person of worth, at least on an equal plane with others.
8. I wish I could have more respect for myself (R)
9. All in all, I am inclined to feel that I am a failure (R)
10. I take a positive attitude toward myself.

Next Steps

- Use θ estimates in future analyses instead of summed scores

```
SelfEsteem <- theta2PL[1,]
```

- Investigate item *and* category functioning using polytomous IRT models

Conclusion

- IRT has clear benefits compared to CTT
- Analysis of scales using IRT models are becoming more accessible and user-friendly
- Improved precision of person estimates for use in future analyses