PSYC 654: Psychometrics and Survey Design
by Lawrence Liu

Task 1: IRT 2PL

1. Discrimination parameter & difficulty parameter:

	Difficulty	Discrimination
Item 1	-0.686	1.336
Item 2	0.711	1.626
Item 3	1.261	1.372
Item 4	-1.462	1.798
Item 5	-1.343	1.157
Item 6	-0.684	1.932
Item 7	0.148	1.423
Item 8	0.051	1.286
Item 9	1.488	1.747
Item 10	-0.103	1.148

1-1. Identifying the easiest and most difficult items:

Using the table above, we will attempt to identify the easiest and most difficult items.
The difficulty parameter (parameter b) mostly ranges from -3 to 3, with -3 indicating the easiest and 3 indicating the most difficult. Therefore, the most difficult item should have a difficulty value closest to 3, while the easiest item should have a value closest to -3. By examining the "Difficulty" column on the above table, we can identify item 9 (highlighted in yellow) is the most difficult item with item 4 (highlighted in cyan) being the easiest item.

2. Picking unfit test takers:

By using the "which(abs(p.fit)>2.0)" code and the irtoys package in RStudio, the following respondents are identified as unfit test takers:
Respondent #: 67 105 115 125 133 143 156 205 279 283 371 442

4. Item fit for all items:

	Statistic	DF	P-value
Item 1	17.58673684	7.00000000	0.01398041
Item 2	17.75243645	7.00000000	0.01313831
Item 3	20.322298678	7.00000000	0.00491402
Item 4	18.830987666	7.00000000	0.008733602
Item 5	6.7478881	7.00000000	0.4555933
Item 6	3.106109e+01	7.00000000	6.058127e-05
Item 7	16.04773472	7.00000000	0.02468381
Item 8	10.2504367	7.00000000	0.1748201
Item 9	13.19718817	7.00000000	0.06744722
Item 10	2.717253e+01	7.00000000	3.102692e-04

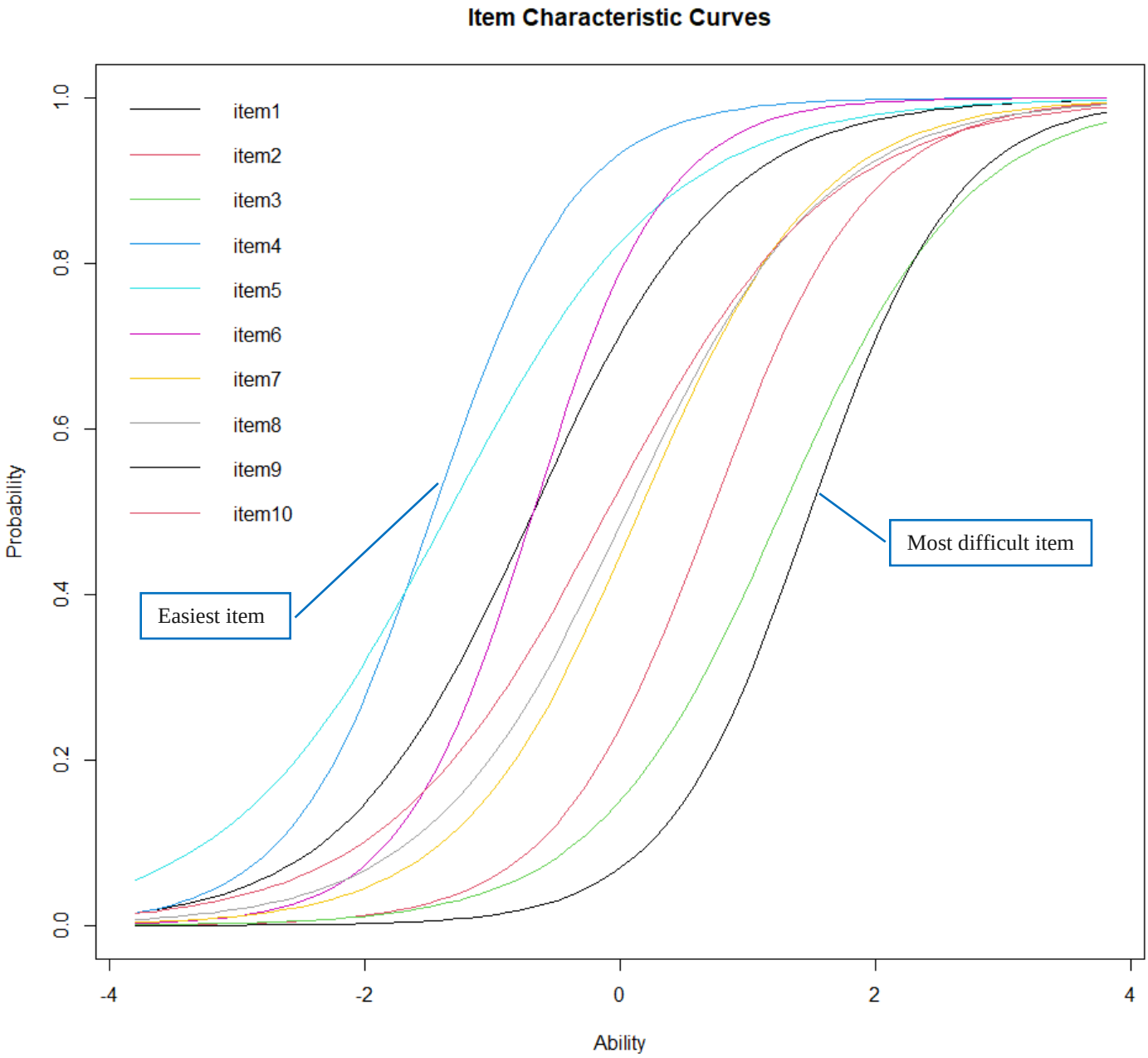
By using the "itf" function in the irtoys package in RStudio, we are able to generate item fit statistics for all 10 items. Looking at the item fit table above, we could identify good items by examining the p-values of individual items. A p-value greater than 0.05 would indicate a good item fit. According to the table above, Items 5, 8, 9 (highlighted in yellow) all have a p-value that's greater than 0.05, which suggests they have a good item fit. The item fit graphs are provide in Appendix A.

5. Theta level:

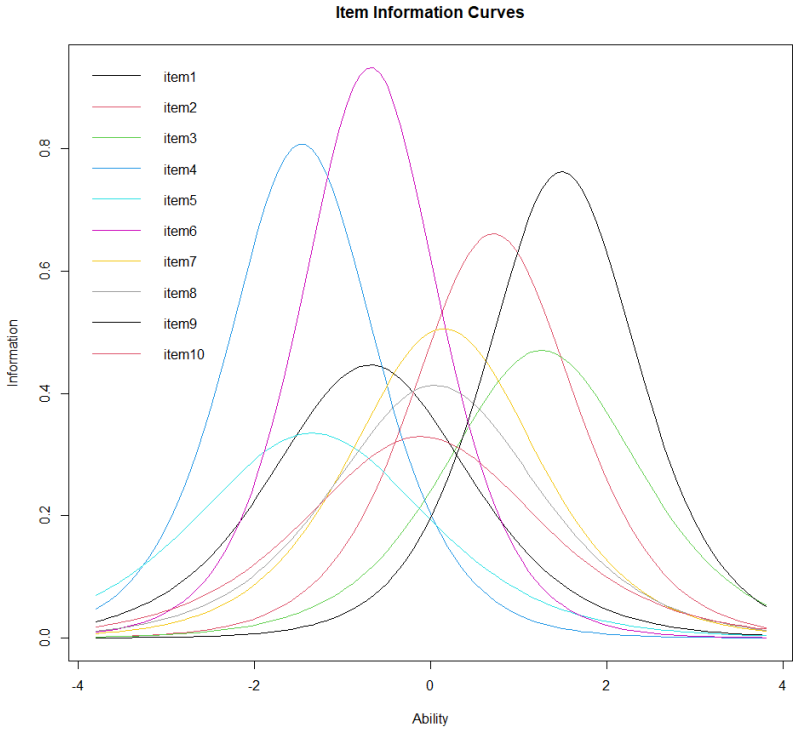
By using the "eap" function in RStudio, we are able to extract the theta (ability level) value of the test takers. The table below would be the descriptive statistics of the theta levels among test takers. Detailed theta level for individual test takers will be included in the .csv file attached with this word document.

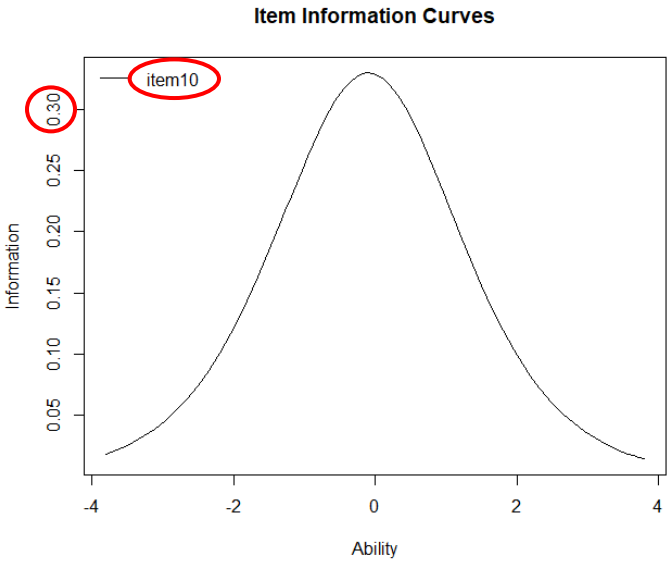
	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
est	1	500	-2.262032e-05	0.86881262	-0.03027479	-0.01340739	0.89666534	-1.922010	1.8769740	3.7989836	0.09856768	-0.4117546	0.038854482
sem	2	500	4.952633e-01	0.03248935	0.48029170	0.48801432	0.01252436	0.471238	0.5983227	0.1270847	1.99037432	3.3640417	0.001452968
n	3	500	1.000000e+01	0.00000000	10.00000000	10.00000000	0.00000000	10.000000	10.0000000	0.0000000	NaN	NaN	0.000000000

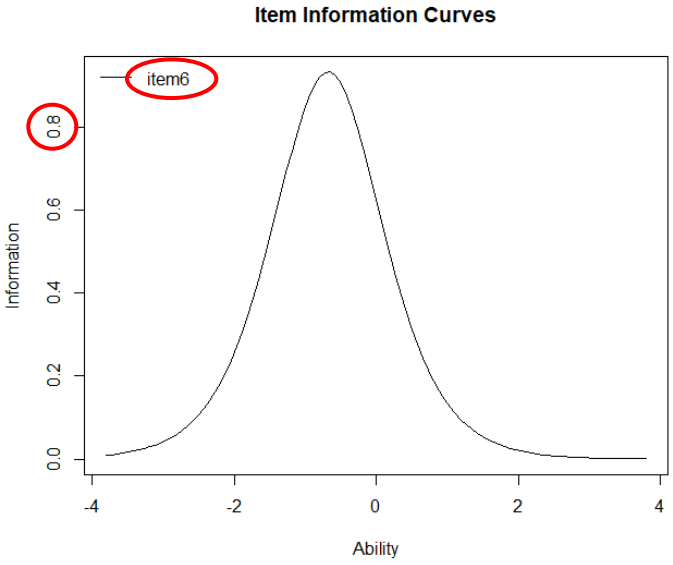
6. Item characteristic curve:



7. Item information curve:







8. Items with the lowest quality and highest quality:

Looking at the item information curves graphs above, we will attempt to identify the item with the highest quality as well as the item with the lowest quality. The item with the highest quality should display the highest item information around the average trait/ability levels and displays less item information at extreme trait/ability levels; the item with the lowest quality should display the opposite. With this information, we could identify that item 10 would be the item with the lowest quality (peaks at just over 0.30 item information), while item 6 has the highest quality (peaks well over 0.80 item information).

8-1. Item that is best to assess people with average ability/ trait levels:

According to the IIC graph above, because item 6 has the most item information around the average trait/ability level, it would be the best item to assess people with average abilities.

Task 2: IRT GRM

1. Discrimination parameter and difficulty parameters:

label_1	Extrmt1	Extrmt2	Extrmt3	Extrmt4	Discrimination
item I1	-3.441	-1.631	0.034	1.631	0.673
item I2	-3.329	-2.332	-1.583	-0.304	1.327
item I11	-2.804	-1.746	-1.055	-0.051	2.522
item I12	-2.732	-1.675	-0.954	-0.048	2.286
item I13	-3.071	-2.084	-1.4	-0.578	1.764
item I14	-3.545	-2.41	-0.961	0.6	1.233
item I15	-3.898	-1.851	-0.913	0.963	1.016
item I17	-2.708	-1.618	-1.065	0.067	1.724
item I8	NON	-2.437	-1.789	-0.62	1.912
item I16	NON	-3.091	-1.67	-0.317	1.755

1-1. Identifying difficult and easy items:

Looking at the table above, we will attempt to examine the level of difficulty/easiness of the selected items. To check the level of difficulty, we will proceed to check the item difficulty parameters. Specifically, we will examine difficulty parameter 4 to help determine the level of difficulty of our items, in other words, we want to examine the trait/ ability level (in our case, trait level, since we are using non-cognitive items) when the probability of respondents to chose "strongly agree" is at 50%. The reason for picking the fourth difficulty parameter is because the default probability of a respondent with an average trait/ability level picking "strongly agree" is the lowest among all four difficulty parameters (when $\theta = 0$, probability of picking higher than "agree" is 0.03). This would help us effectively distinguish which items are difficult, and which are not. As the table above would demonstrate, 6 out of the 10 items (highlighted in yellow) displayed a lower-than-average trait level (assumed negative value) when the probability of respondents choosing "strongly agree" is at 50%. This suggests that 6 out of the 10 items are easy (does not require respondents with high trait level to select). On the other hand, the most difficult (require respondents with high trait level to select) item would be item I1 (highlighted in cyan), with a difficulty parameter value of 1.63, which suggests that it would require a higher-than-average trait level when the probability of respondents choosing "strongly agree" is at 50%.

1-2. Identifying items with good and bad qualities:

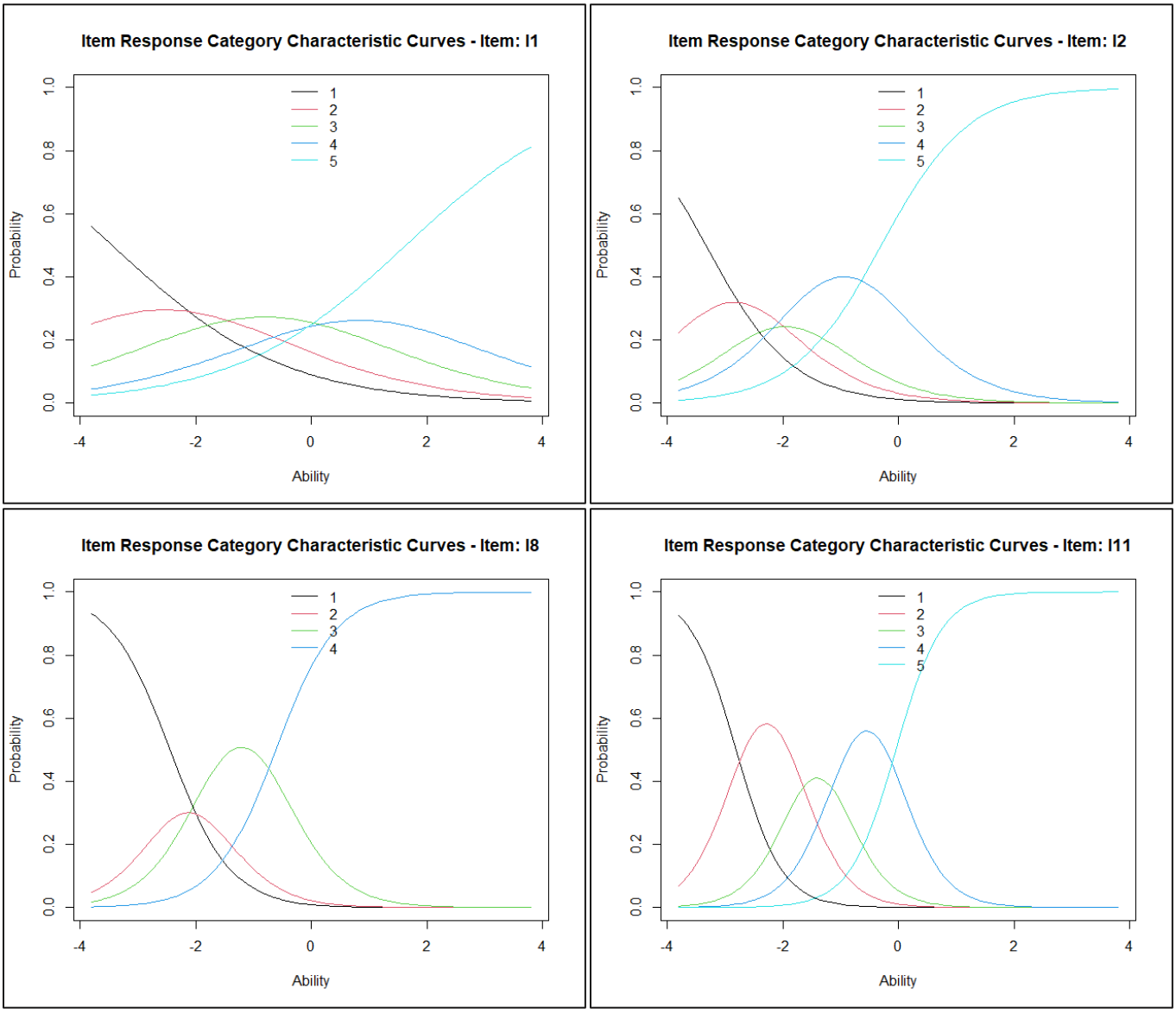
To identify the qualities of our items, we will first examine the discrimination parameter to identify items with a parameter value that's lower than 0.5 (cutoff value). According to the table above, none of the items displayed a discrimination parameter value that's less than 0.5. However, it is noteworthy that item I1 (highlighted in yellow) has the lowest discrimination value out of all items. This might suggest item I1 has a relatively poor quality. Additionally, if we examine the item response category characteristics curve (CCC) of item I1, we can see that the steepness of parameter a is quite gentle compared with the CCCs of other items. This further suggests that item I1 has a poor quality. Other than item I1, the rest of the items all display a good parameter a value, which suggest good item qualities.

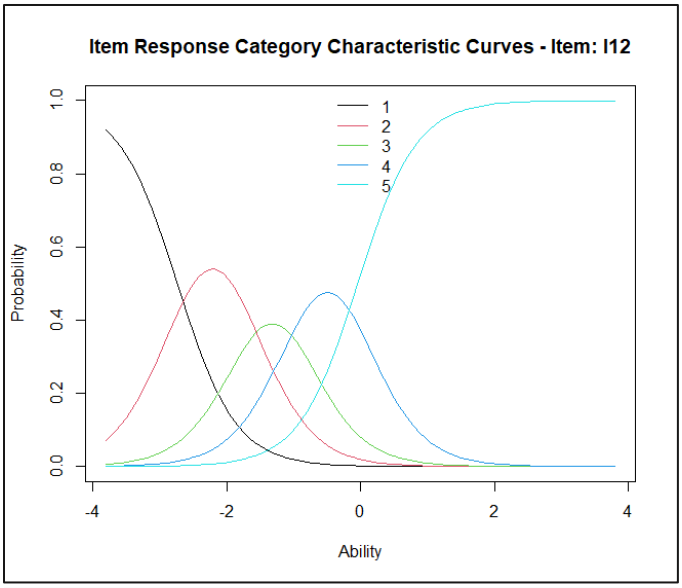
1-3. Ideas for improving my items:

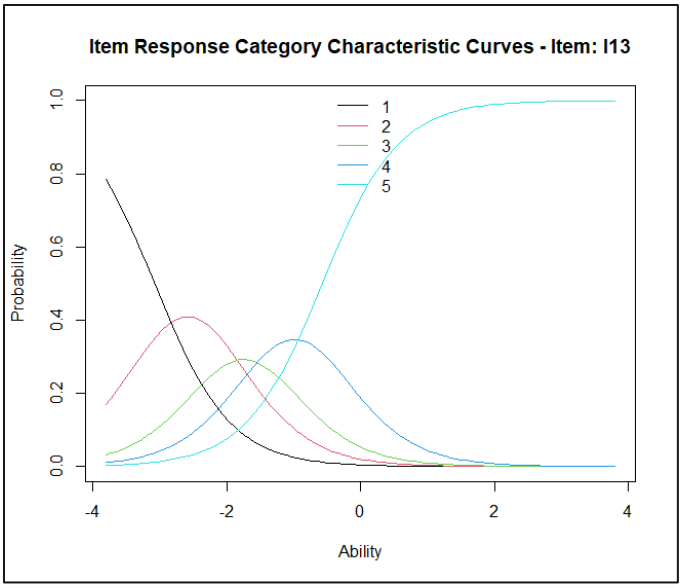
The items that are selected for task 2 are items assessing the level of integrity among respondents. My first idea of revision would be to revise item I1 "*I think that people who take ethical shortcuts are more likely to succeed than those who do not.*" Because it is the item that require the highest trait level to choose "strongly agree" at a 50% probability. In other words, I1 is a "difficult" item. The other reason would be item I1's relatively poor ability to discriminate among respondents. Since this item requires reverse scoring, a logical approach of revising would be rephrasing the item. For example, I could rephrase item I1 into "*I believe that ethical people are more likely to succeed than those who are not.*" My second idea of revision would be looking for a way to elevate the "difficulty" of the entire set of items, since it does not require a respondent to have a higher-than-average trait level to pick "strongly agree" at a 50% probability. Let's use the two items with the lowest difficulty value (items I13 and I16) to experiment. For item I13, I could maybe rephrase "*It is okay to steal from work if it is a minor theft.*" Into "*It is unacceptable to steal paperclips from work even if it is just a minor theft.*" This way we eliminate the need for revers-coding, and use paperclips, a genuinely insignificant item as an example, to modify the sentiment of the item. As for item I16, I could rephrase "*I have a strong moral code.*" Into "*I consistently adhere to a strong more code in all of my actions.*" By changing "*I have"* to "*I consistently adhere to*" might deter respondents with an average integrity level to pick "strongly agree" when answering this item. Additionally, I should

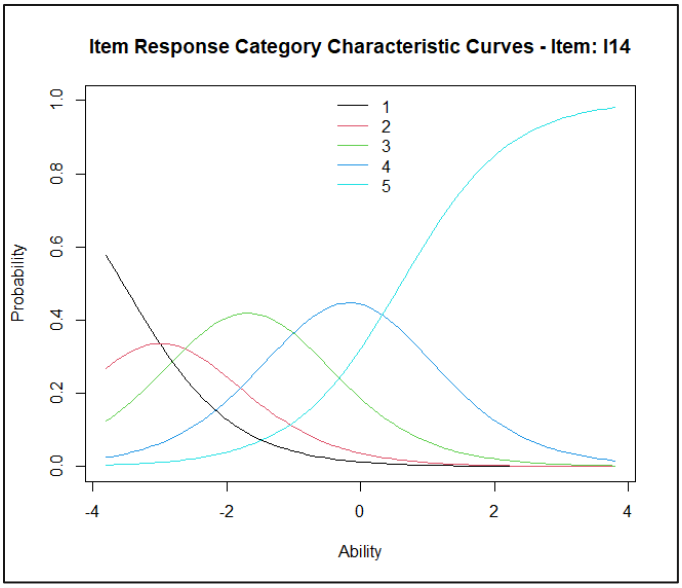
also revise items I8 and I16 since none of the respondents picked "strongly disagree" for these items (highlighted in green in the above table). I've already provided a revision idea for I16, and here is how I could fix I8: I could rephrase I8 from "*I believe honesty is an integral part of my job*." Into "*I strongly believe that being 100% honest is an integral part of my job*." By modifying "believe" with the word "strongly," as well as adding "100%" to the item statement could potentially motivate respondents with lower-than-average integrity levels or respondents that have an occupation that requires occasional "lying" (for example, salespeople tend to use a little bit of white lies here and there to pamper their clients, etc.) to pick "strongly disagree" as their response. Lastly, since this set of items are designed to measure integrity levels, a non-cognitive trait, we should also consider the necessity of making the items "difficult" and "discriminate" across the board. It will ultimately be dependent on the use of these items. For instance, if these items are used to assess the integrity level of court officials (e.g., court judges), it might be expected that all of the respondents will score very high (picking "strongly agree" for most items), and thus lowering the discriminability of the items.

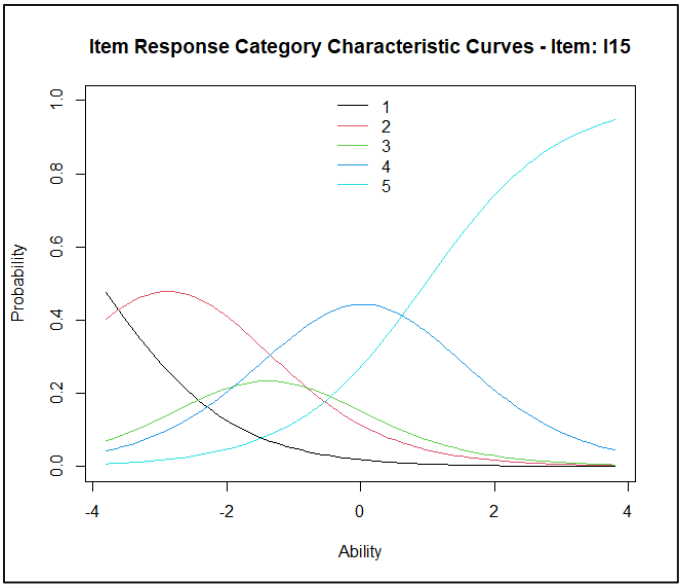
2. Category characteristic curves:

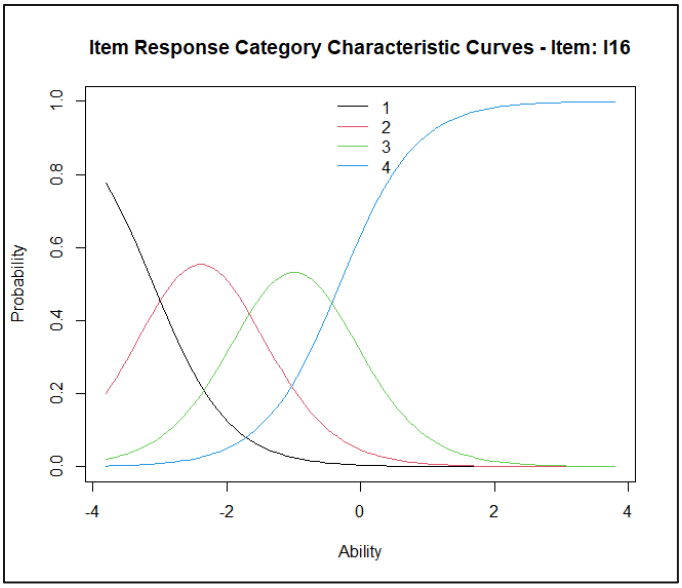


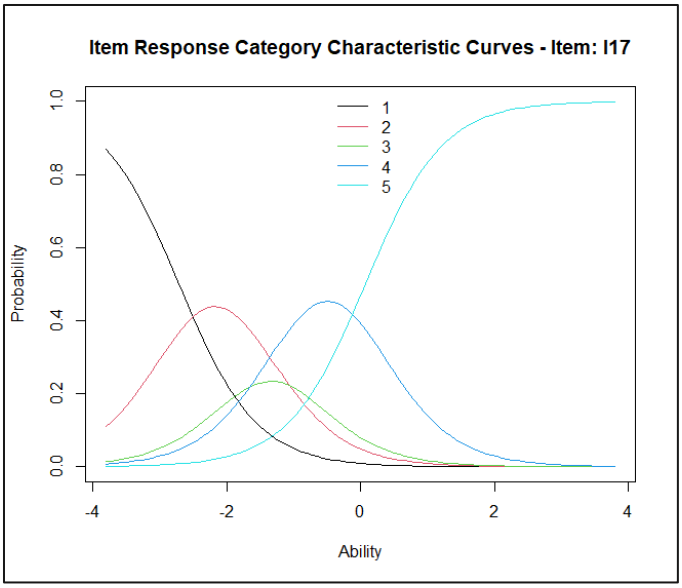

Item Response Category Characteristic Curves - Item: I12
Probability
Ability
1
2
3
4
5


Item Response Category Characteristic Curves - Item: I13
Probability
Ability
1
2
3
4
5

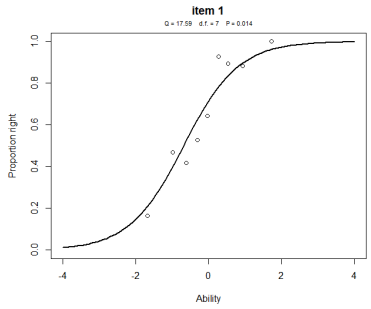

Item Response Category Characteristic Curves - Item: I14
Probability
Ability
1
2
3
4
5

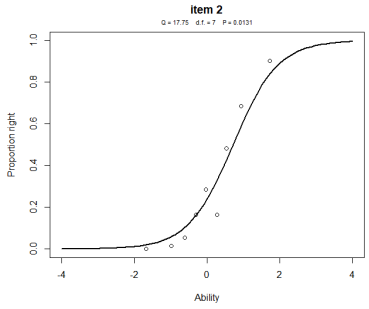

Item Response Category Characteristic Curves - Item: I15
Probability
Ability
1
2
3
4
5

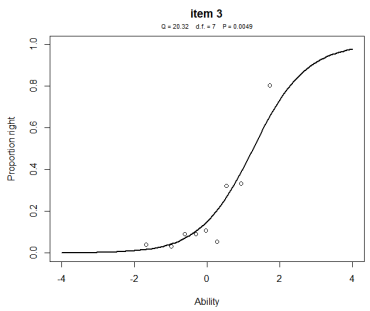

Item Response Category Characteristic Curves - Item: I16
Probability
Ability
1
2
3
4

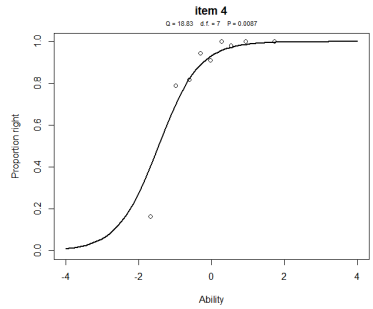

Item Response Category Characteristic Curves - Item: I17
Probability
Ability
1
2
3
4
5

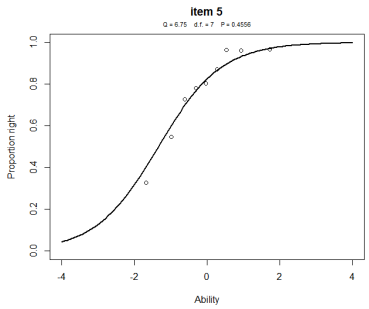
Appendix A – Item Fit Graphs

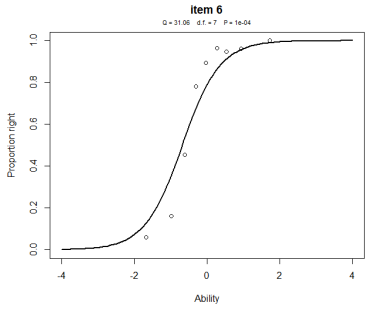


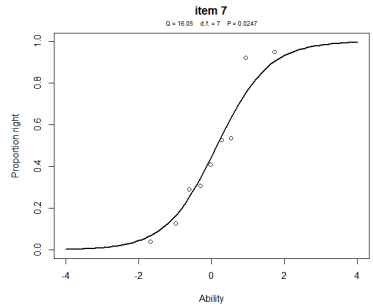


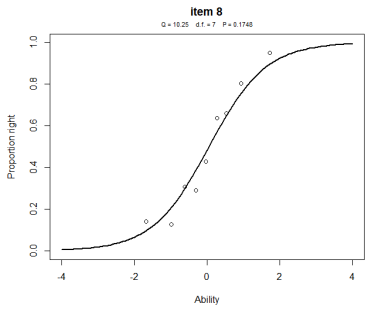


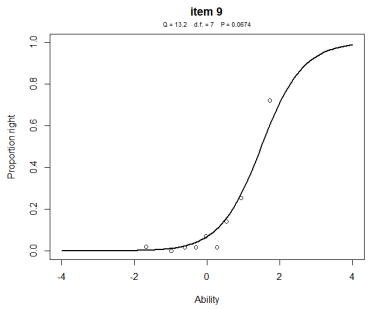


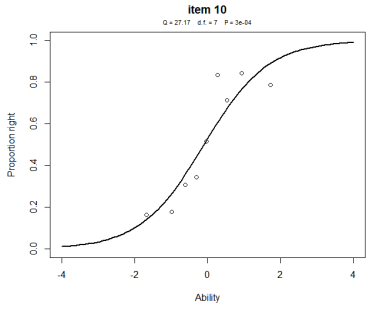












Appendix B – R code for Task 1

```
## manually set working directory

## install "ltm" package

install.packages("ltm")
install.packages("irtoys")

## load "ltm" and "psych"

library("ltm")
library("irtoys")
library("psych")
library("formattable")

dat1 = read.csv(file.choose(), header=T)              ## Import data


## Basic statistics using ltm
## Focus on percentages of correct answers by questions but you can confirm Point Biserial correlation with Total Score and alpha too

descript(dat1)


## Examine uni-diemnsionality

polychoric.cor <- polychoric(dat1)          ## confirm ploychoric correlation

print(polychoric.cor$rho)                  ## see the result of ploychoric correlation

eigen(polychoric.cor$rho)$value            ## To check uni-dimensionality, check eigenvalues

VSS.scree(polychoric.cor$rho,main="scree plot")    ## depict the scree plot


## Estimation of IRT parameters with 2PL
# use ltm package

para_result <- ltm(dat1 ~ z1, IRT.param = TRUE)

para_result     ## To see results, write the save data name ("para_result" in this code)

coef(para_result)   ## When you want to see only parameter estimates

summary(para_result)   ## When you want to see all indecies

# use irtoys package

para_result2 <- est(resp=dat1, model="2PL", engine="ltm")
ip <- para_result2$est

describe(ip)                ## summary of parameters

p.fit <- api(dat1, ip)        ## person fit
```

```
p.fit

which(abs(p.fit)>2.0)        ## pick up the persons which are not good fit

itf(dat1, ip, item=1, main=paste0("item 1"))   ## item fit (can check local independence
itf(dat1, ip, item=2, main=paste0("item 2"))
itf(dat1, ip, item=3, main=paste0("item 3"))
itf(dat1, ip, item=4, main=paste0("item 4"))
itf(dat1, ip, item=5, main=paste0("item 5"))
itf(dat1, ip, item=6, main=paste0("item 6"))
itf(dat1, ip, item=7, main=paste0("item 7"))
itf(dat1, ip, item=8, main=paste0("item 8"))
itf(dat1, ip, item=9, main=paste0("item 9"))
itf(dat1, ip, item=10, main=paste0("item 10"))

## p>.05 indicates good fit

theta <- eap(resp=dat1, ip=ip, qu=normal.qu())

thetadescrip <- describe(theta [,])

formattable(thetadescrip)

write.csv(theta, "abilitylevel.csv", quote=FALSE)

## Depict ICC, IIC, & TIC

#ltm package

plot(para_result, type = "ICC", legend = TRUE)  # ICC for all items
plot(para_result, type = "ICC", legend = TRUE, items=4) # ICC for an individual item
plot(para_result, type = "IIC", legend = TRUE)  # IIC for all items
plot(para_result, type = "IIC", legend = TRUE, items=6) # IIC for an individual item
plot(para_result, type = "IIC", items=0)  # TIC

#irtoys package
plot(trf(ip), co=NA, label=True)                    #ICC
plot(iif(ip), co=NA, label=True)                    #IIC
plot(tif(ip), co=NA, label=True)
```

Appendix C – R code for Task 2

```
## Manually set working directory

## load "ltm"

library("ltm")
library("psych")
library(tidyverse)

dat1 = read.csv(file.choose(), header=T)              ## Import data

## if your data include reverse items, do reverse coding

dat1[,c(1,6)] <- 6-dat1[,c(1,6)]    ## reverse code items I1 & I13

dat1    ## check if reverse coded

## Examine uni-diemnsionality

cor_result <- cor(dat1, use="complete.obs")          ## confirm ploychoric correlation

eigen(cor_result)$value                          ## To check uni-dimensionality, check eigenvalues

VSS.scree(cor_result,main="scree plot")    ## depict the scree plot

## Basic statistics using ltm
## Focus on percentages of options

descript(dat1)     ## If there are items without any options, need to take care of them

## if you find items with an option that no one selected and get an error in R
## one solution as Huub Hoofs suggested is to subtract 1 from the variable(s) that have counts of 0
```

```
dat2 <- within(dat1, I8 <- (I8 - 1))
dat3 <- within(dat2, I16 <- (I16 - 1))

descript(dat3)

## Estimation of IRT parameters with GRM

para_grm <- grm(dat3, IRT.para=T)   ## estimate parameters
para_grm                        ## To see results, write the save data name ("para_grm" in this code)
coef(para_grm)                    ## When you want to see only parameter estimates
summary(para_grm)                 ## When you want to see all indecies

## Export the parameters

#when there is/are item(s) with an option that no one selected
result <- coef(para_grm)

result1<- rbind(result$I1, result$I2, result$I11, result$I12, result$I13, result$I14, result$I15, result$I17)
label_1<-c("item I1", "item I2", "item I11", "item I12", "item I13", "item I14", "item I15", "item I17")
result2<-cbind(label_1, result1)

result3<- rbind(result$I8, result$I16)
label_2<-c("item I8", "item I16")
emp_col <- rep("NON", 2)
result4<-cbind (label_2, emp_col, result3)

final<-rbind(result2, result4)

write.csv(final,"task2_result.csv")      ## export the result

## Depict IRCCC

plot(para_grm, type = "ICC", legend = TRUE, items=1) # IRCCC for item 1
```

```
plot(para_grm, type = "ICC", legend = TRUE, items=2) # IRCCC for item 2
plot(para_grm, type = "ICC", legend = TRUE, items=3) # IRCCC for item 3
plot(para_grm, type = "ICC", legend = TRUE, items=4) # IRCCC for item 4
plot(para_grm, type = "ICC", legend = TRUE, items=5) # IRCCC for item 5
plot(para_grm, type = "ICC", legend = TRUE, items=6) # IRCCC for item 6
plot(para_grm, type = "ICC", legend = TRUE, items=7) # IRCCC for item 7
plot(para_grm, type = "ICC", legend = TRUE, items=8) # IRCCC for item 8
plot(para_grm, type = "ICC", legend = TRUE, items=9) # IRCCC for item 9
plot(para_grm, type = "ICC", legend = TRUE, items=10) # IRCCC for item 10
```