



Computing

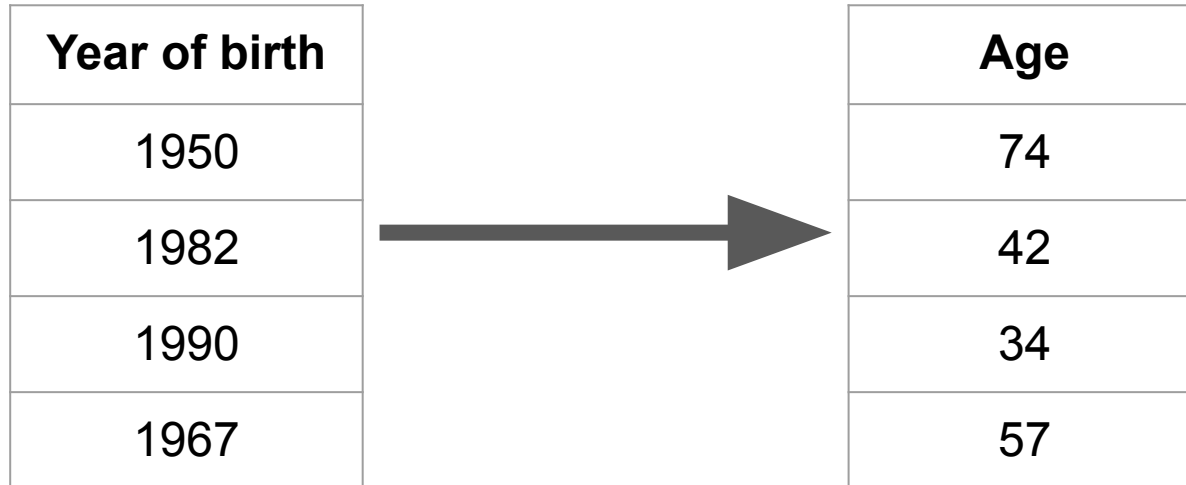
Tolga Tezcan, PhD

Learning outcomes

- Learn how to compute new variables
- Learn how to create index variables
- Learn common computing mistakes

Computing: why?

Sometimes we need to compute a new variable based on existing information (from other variables) in our data.



Index

- An **index** is an accumulation of scores from a variety of individual variables.
- It is difficult to measure social issues with simply one variable (question).
- Instead, we can use several different variables (questions) that deal with the social issue and create an index of the included variables.



Items

Think about your **ethnicity/race**. What group do you belong to? **Do you think of yourself as:** Asian? Black? Latino? White? Native American? American? Caribbean? Irish? Italian? Korean? Another group?

Your Ethnicity/Race: _____

How often have any of the things listed below ever happened to you, **because of your ethnicity?**

BECAUSE OF YOUR ETHNICITY/RACE...

A. How often . . .

1. Have you been treated unfairly by teachers, principals, or other staff at school?
2. Have others thought you couldn't do things or handle a job?
3. Have others threatened to hurt you (ex: said they would hit you)?
4. Have others actually hurt you or tried to hurt you (ex: kicked or hit you)?
5. Have policeman or security officers been unfair to you?
6. Have others threatened to damage your property?
7. Have others actually damaged your property?
8. Have others made you feel like an outsider who doesn't fit in because of your dress, speech, or other characteristics related to your ethnicity?
9. Have you been treated unfairly by co-workers or classmates?
10. Have others hinted that you are dishonest or can't be trusted?
11. Have people been nice to your face, but said bad things about you behind your back?
12. Have people who speak a different language made you feel like an outsider?
13. Have others ignored you or not paid attention to you?
14. Has your boss or supervisor been unfair to you?
15. Have others hinted that you must not be clean?
16. Have people not trusted you?
17. Has it been hinted that you must be lazy?

Never	Rarely	Occasionally	Frequently	Always
1	2	3	4	5

[Calculate Perceived Ethnic Discrimination Index Score](#)

Index of happiness



1) **happy** – general happiness

1: very happy; 2: pretty happy; 3: not too happy

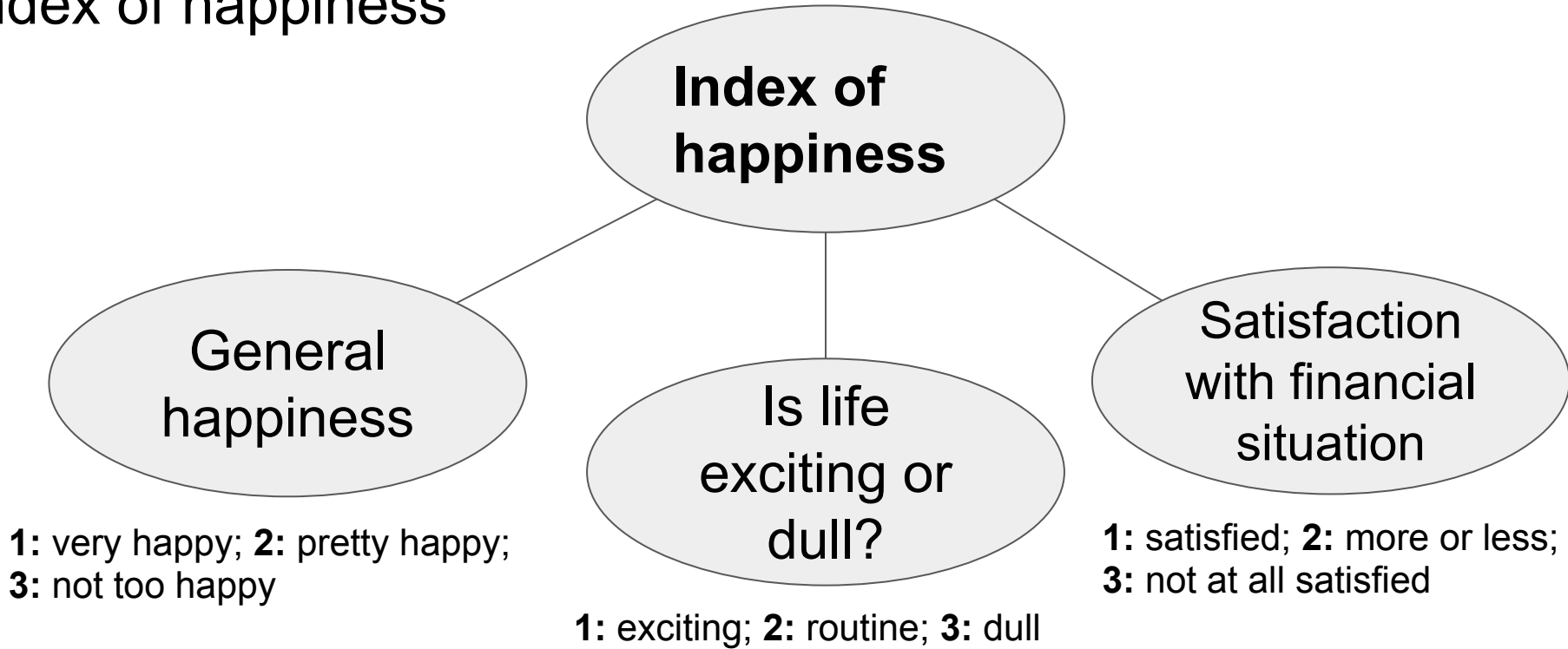
2) **life** – is life exciting and dull

1: exciting; 2: routine; 3: dull

3) **satfin** – satisfaction with financial situation

1: satisfied; 2: more or less; 3: not at all satisfied

Index of happiness



Index of Happiness = the mean score of (happy + life + satfin)

Index of happiness - steps

- (1) Check the frequency distribution of individual variables**
- (2) Recode variables, if necessary**
- (3) Create an index variable (computation)**
- (4) Check the distribution of your index variable**

Index of happiness - steps

(1) Check frequency distribution of individual variables

We always check the distribution of the variables that we want to use in our analyses (to see what **1, 2, 3** mean). Frequency for categorical; descriptives for continuous variables

general happiness (x) <numeric>

<i>val</i>	<i>label</i>	<i>freq</i>	<i>raw.prc</i>	<i>valid.prc</i>	<i>cum.prc</i>
1	very happy	779	21.98	22.13	22.13
2	pretty happy	1942	54.80	55.17	77.30
3	not too happy	799	22.55	22.70	100.00
NA	NA	24	0.68	NA	NA

```
freq(gss$happy, out = "v")
```


```
freq(gss$life, out = "v")
```

```
freq(gss$satfin, out = "v")
```

Index of happiness

From 1 to 3, the happiness decreases. Variables are coded so high value (3) indicates less happiness. **We need to fix this!**

happy	more or less	Not happy
1	2	3



happy	more or less	Not happy
3	2	1

Index of happiness - steps

2) Recode variables, if necessary

happy: the original variable we want to recode.

happynew: new name for the recoded variable. This will be added to GSS dataset.

very happy, **pretty happy** and **not too happy**: labels for the categories. These will appear on the table.

From 1 to 3, the happiness decreases. Variables are coded so high value (3) indicates less happiness. 1s should be 3, 2s should be 2, and 3s should be 1

```
gss$happynew <- rec(gss$happy, rec =  
"1=3 [very happy];  
2=2 [pretty happy];  
3=1 [not too happy]", append = FALSE)
```

```
gss$satfinnew <- rec(gss$satfin, rec =  
"1=3 [satisfied];  
2=2 [more or less];  
3=1 [not at all]", append = FALSE)
```

```
gss$lifnew <- rec(gss$life, rec =  
"1=3 [exciting];  
2=2 [routine];  
3=1 [dull]", append = FALSE)
```

Index of happiness - steps

3) Create an index variable

```
gss <- gss %>%  
  rowwise() %>%  
  mutate (hapindex = mean (c(happynew, lifenew, satfinnew)))
```

Index of happiness - steps

4) Check the distribution of your index variable

```
descr(gss$hapindex, out = "v", show = "short")
```

<i>Variable</i>	<i>N</i>	<i>Missings (%)</i>	<i>Mean</i>	<i>SD</i>
dd	2309	34.85	2.10	0.47

The happiness index score of the GSS respondents is **2.10** out of 3, with standard deviation 0.47.

Overview

happy	life	satfin	happynew	lifeneu	satfinnew	hapindex
1	2	2	3	2	2	2.33
2	3	2	2	1	2	1.66
1	2	3	3	2	1	2.00
3	3	1	1	1	3	1.66

```
gss$satfinnew <- rec(gss$satfin, rec =  
"1=3 [exciting];  
2=2 [routine];  
3=1 [dull]", append = FALSE)
```

```
gss <- gss %>%  
  rowwise() %>%  
  mutate (hapindex =  
  mean (c(happynew,lifeneu,satfinnew)))
```

Common computing mistakes (1)

Not using the new (recoded) variables in computation code

```
gss <- gss %>%  
  rowwise() %>%  
  mutate (hapindex = mean (c(happy,life,satfin)))
```



```
gss <- gss %>%  
  rowwise() %>%  
  mutate (hapindex = mean (c(happynew,lifenew,satfinnew)))
```



Common computing mistakes (2)

When we compute variables and create an index, the new (computed) variable is **continuous**.

It becomes **CONTINUOUS** because we have created a score, and we treat it as a real number.

Therefore, we use the **DESCR** code to see the distribution (mean and standard deviation)

```
frq(gss$hapindex, out = "v", show = "short")
```



```
descr(gss$hapindex, out = "v")
```



Common computing mistakes (3)

Not using a model code

Use the [Code templates](#) page.

computing 1

```
gss <- gss %>%  
  rowwise() %>%  
  mutate (parentseducmean2 = mean (c(maeduc,paeduc)))
```

computing 2 (with recoding)

```
frq(gss$happy, out = "v")  
  
gss$happynew <- rec(gss$happy, rec =  
  "1=3 [very happy];  
  2=2 [pretty happy];  
  3=1 [not too happy]", append = FALSE)  
  
frq(gss$life, out = "v")  
  
gss$lifefew <- rec(gss$life, rec =  
  "1=3 [exciting];  
  2=2 [routine];  
  3=1 [dull]", append = FALSE)  
  
frq(gss$atfin, out = "v")  
  
gss$atfinnew <- rec(gss$atfin, rec =  
  "1=3 [satisfied];  
  2=2 [more or less];  
  3=1 [not at all]", append = FALSE)  
  
gss <- gss %>%  
  rowwise() %>%  
  mutate (hapindex = mean (c(happynew,lifefew,atfinnew)))
```

computing 3 (with recoding)

```
frq(gss$socrel, out = "v")  
  
gss$socrelnew <- rec(gss$socrel, rec =  
  "1=7 [almost daily];  
  2=6 [once or twice a week];  
  3=5 [several times a month];
```

Why can't I see a table when I compute a variable?

When you compute a variable, **you will not see a table**. That code will create a new variable.

If you want to see the distribution of your new (computed) variables:

- Descriptive code will generate a descriptive table (for continuous variables):

```
descr(gss$hapindex, out = "v", show = "short")
```