

Part I. (9 points)

(6^{pts}) **1.** In-class or take-home.

(a) (1 pt) Name:

(b) (1 pt) What would you like to be called:

(c) (1 pt) Degree and major:

(d) (1 pt) Do you have a job? If so, where do you work and what do you do?

(e) (1 pt) Why are you taking this course?

(f) (1 pt) What is something about you that is probably not true of other students in the class?
(for example, an unusual experience, hobby, skill, or interest):

(3^{pts}) **2.** In-class questions

(a) (1 pt) Learning outcomes, syllabus:

Circle the syllabus learning outcomes that are important for you to achieve:

1 2 3 4 5 6 7 8 9 10 11 12 13

(b) (1 pt) Check mark the one that is *most* important to you.

(c) (1 pt) Learning outcomes, general

Mark the number:

1 2 3

Solution: Results of Fall 2012:

```
# ADA1 Stat 427/527 Fall 2012
# 8/21/2012 6:24PM
# learning outcomes from HW00
```

```
# With so many values to enter by hand, I needed to come up with
# a quick and easy way of getting the data into R.
# I decided to type all the values from each sheet on a row,
# originally I used scan() for direct copy/paste into to R console.
# The most important outcome is multiplied 100 so it's easy to code these later.
```

```
outcomes <- c(
  1, 3, 7, 8, 9, 11, 13, 900
, 1, 3, 7, 8, 11, 13, 1100
, 1, 3, 5, 6, 7, 10, 12, 13, 300
, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13
, 1, 2, 3, 5, 6, 7, 8, 11, 12, 13, 600
, 3, 6, 8, 10, 11, 12, 600
, 1, 6, 8, 11, 800
, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 800
, 6, 8
, 1, 3, 4, 8, 100
, 6, 8, 600
, 6, 8, 600
, 1, 4, 5, 6, 7, 8, 10, 11, 13, 600
, 1, 6, 8, 100
, 3, 8, 10, 11, 12, 1100
, 1, 3, 5, 6, 7, 8, 12, 13, 1300
```

```
, 1, 3, 6, 7, 8, 10, 11, 12, 1200
, 3, 4, 6, 7, 8, 10, 11, 13, 800
, 3, 8, 11, 13, 800
, 3, 6, 8, 11, 12, 13, 800, 1100
, 3, 7, 8, 9, 10, 11, 12, 800
, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 800
, 1, 2, 4, 10, 11, 1000
, 1, 3, 6, 8, 12, 100
, 1, 7, 8, 12
, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 900
, 2, 6, 7, 10, 13, 1000
, 7
, 1, 6, 8, 10, 800
, 1, 3, 6, 8, 9, 13, 300
, 3, 4, 6, 7, 8, 10, 12, 13, 1300
, 1, 2, 5, 7, 10, 13, 100
, 11, 12, 13, 1200
, 1, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 300
, 1, 2, 3, 7, 9, 100
, 3, 4, 6, 7, 10, 11, 600
, 4, 7, 8, 11, 13
, 2, 4, 7, 8, 12, 800
, 2, 3, 7, 8, 9, 10, 12, 13, 1300
, 8, 9, 10, 11, 12, 13, 1200
, 1, 2, 6, 7, 8, 11, 700
, 1, 6, 7, 11, 600
, 6, 7
, 1, 2, 3, 4, 6, 8, 10, 12, 13, 800
, 4, 7, 8, 10, 11, 12, 13, 400
, 1, 3, 4, 6, 8, 9, 11, 600
, 4, 5, 6, 8, 10, 11, 12, 13, 400
, 1, 2, 3, 6, 8, 9, 11, 12, 13
, 1, 8, 11, 800
, 1, 3, 4, 6, 7, 8, 9, 10, 600
, 6, 8, 12, 13, 800
, 1, 3, 6, 7, 9, 11, 12, 300
, 2, 3, 5, 10, 1000
, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 1100
, 1, 3, 6, 8, 9, 12, 900
, 1, 3, 6, 7, 8, 9, 10, 13, 600
, 3, 6, 8, 9, 300
, 1, 4, 6, 8
, 1, 2, 3, 6, 8, 11, 12, 100
, 1, 3, 5, 6, 8, 9, 10, 11, 12, 13, 800
, 2, 3, 7, 11, 1100
, 3, 4, 6, 7, 8, 12, 13, 1300
, 1, 6, 8, 9, 10, 11, 12, 13, 600
, 1, 3, 7, 8, 11, 300
, 2, 3, 6, 7, 8, 11, 300
, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 600
, 1, 4, 6, 8, 13
, 1, 2, 6, 9, 100
, 6, 7, 700
, 1, 4, 8, 13, 800
, 1, 4, 6, 8, 10, 11, 13, 800
, 2, 3, 9, 11, 12, 13, 1300
, 1, 3, 8, 9, 11, 13, 900
, 1, 2, 3, 4, 7, 8, 13, 1300
, 1, 4, 6, 8, 10, 12, 600, 800
, 1, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 400, 800
, 1, 4, 6, 7, 9, 13, 600
, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 1100
, 1, 8, 9, 11, 12, 13, 1100
, 7, 11, 1100
, 2, 4, 7, 8, 10, 11, 200
, 1, 3, 4, 6, 7, 8, 10, 11, 12, 600
, 1, 6, 8
```

```

, 1, 4, 6, 8, 13, 1300
, 2, 13, 1300
, 3, 7, 8, 10, 11, 13, 800
, 3, 4, 6, 8, 9, 10, 12, 600
, 1, 2, 3, 6, 7, 8, 10, 12, 13, 300
, 3, 4, 8, 800
, 1, 3, 6, 8, 11, 13, 800
, 1, 3, 8, 13, 100
, 1, 4, 7, 8, 13, 1300
, 3, 6, 8, 10, 11, 300
, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 1100
)

# end scan(), need one blank line after last data value

# initialize type "all" or "most" with NA (not available)
type <- rep(NA, length(outcomes));
# create a data.frame df.12 with both outcomes and type
df.12 <- data.frame(outcomes, type)
# assign type to "all" if outcome is < 100
df.12$type[(df.12$outcomes < 100)] <- "all"
# assign type to "most" if outcome is >= 100
df.12$type[(df.12$outcomes >= 100)] <- "most"
# divide the most outcome numbers by 100 to put on original scale
df.12$outcomes[(df.12$type == "most")] <- df.12$outcomes[(df.12$type == "most")]/100

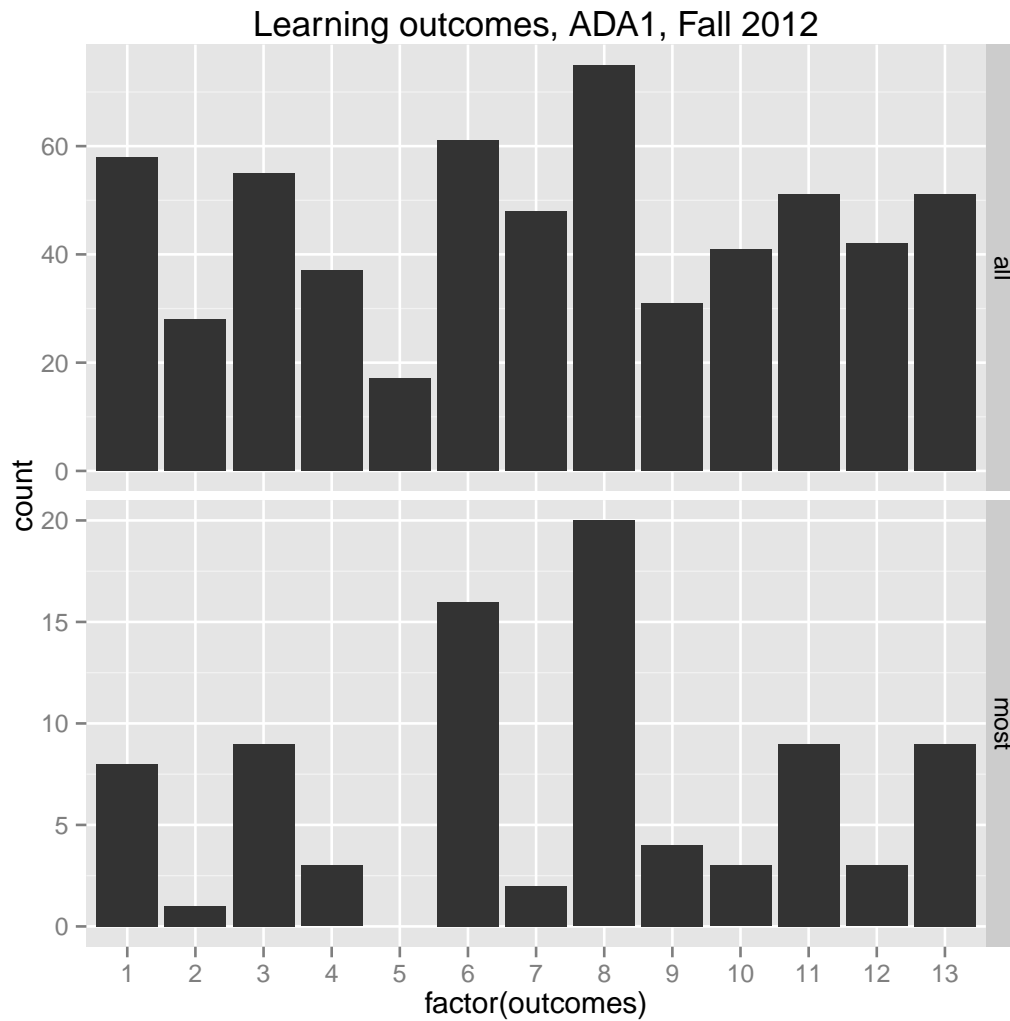
# tabular results
xtabs( ~ outcomes + type, data = df.12)

##           type
## outcomes all most
##      1    58    8
##      2    28    1
##      3    55    9
##      4    37    3
##      5    17    0
##      6    61   16
##      7    48    2
##      8    75   20
##      9    31    4
##     10    41    3
##     11    51    9
##     12    42    3
##     13    51    9

# load the ggplot2 plotting library
library(ggplot2)
# init plot where outcomes are on the horizontal axis
p <- ggplot(df.12, aes(x = factor(outcomes)))
# add bars
p <- p + geom_bar()
# separate the plots by type "all" and "most", let vertical scale vary per plot
p <- p + facet_grid(type ~ ., scales = "free")
# Title the plot
p <- p + labs(title = "Learning outcomes, ADA1, Fall 2012")
# Print the plot
print(p)

#ggsave("C:/Dropbox/UNM/teach/ADA1_stat527/assess/ADA1_HW_00_F12_plot.png", p, height = 6, width = 8)

```



Results of Fall 2013:

```
# ADA1 Stat 427/527 Fall 2013
# 8/20/2013 12:51PM
# learning outcomes from HW00
```

```
# With so many values to enter by hand, I needed to come up with
# a quick and easy way of getting the data into R.
# I decided to type all the values from each sheet on a row,
# originally I used scan() for direct copy/paste into to R console.
# The most important outcome is multiplied 100 so it's easy to code these later.
```

```
outcomes <- c(
  3, 4, 7, 8, 10, 11, 12, 1000
, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 100
, 2, 3, 7, 8, 11, 800
, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 400
, 2, 4, 6, 8, 9, 10, 11, 12, 13, 1300
, 1, 4, 6, 8, 400
, 1, 3, 4, 9, 12, 13, 400
, 1, 3, 4, 6, 8, 9, 11, 12, 13, 1300
, 1, 3, 6, 10, 13, 100
, 6, 10, 1000
, 1, 2, 4, 6, 7, 8, 10, 11, 13, 800
, 1, 3, 4, 7, 8, 9, 10, 11, 13, 1100
```

```

, 1, 2, 3, 4, 6, 8, 9, 10, 11, 12, 13, 100
, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 100
, 1, 3, 4, 6, 7, 9, 10, 11, 12, 13, 100
, 1, 2, 4, 6, 7, 8, 9, 10, 11, 12, 600
, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 100
, 1, 2, 3, 4, 6, 7, 9, 11, 12, 13, 700
, 1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 400
, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 600
, 1, 2, 3, 4, 7, 9, 11, 12, 13, 1100
, 1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 300
, 1, 2, 3, 6, 7, 8, 9, 10, 11, 12, 13, 900
, 1, 2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 600
, 2, 3, 5, 7, 9, 12, 13, 300
, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 400
, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 800
, 1, 2, 3, 4, 5, 10, 11, 12, 13, 1100
, 1, 3, 4, 6, 8, 10, 11, 13, 100
, 1, 2, 3, 4, 6, 8, 10, 11, 12, 13, 800
, 1, 4, 6, 8, 10, 13, 800
, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 400
, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 800
, 1, 2, 3, 5, 7, 8, 11, 12, 100
, 1, 3, 8, 13, 800
, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 1100
, 1, 3, 4, 5, 6, 8, 11, 13, 800
, 1, 2, 3, 4, 6, 7, 8, 9, 11, 12, 13, 100
, 1, 2, 3, 4, 6, 8, 11, 12, 13, 800
, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 1200
, 1, 3, 4, 6, 7, 8, 9, 11, 12, 700
, 1, 3, 5, 6, 8, 10, 11, 13, 100
, 3, 4, 6, 7, 8, 9, 10, 11, 13, 800
, 1, 2, 3, 6, 7, 8, 12, 800
, 1, 4, 5, 6, 8, 9, 10, 11, 12, 13, 800
, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 100
, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12, 13, 400
, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 700
, 1, 3, 4, 6, 7, 8, 9, 11, 12, 600
, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 1200
, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 100
, 8, 9, 10, 11, 13, 800
, 1, 4, 6, 8, 9, 10, 12, 600
, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 800
, 1, 6, 7, 8, 10, 600
, 2, 4, 6, 7, 8, 10, 11, 1000
, 1, 2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 1300
, 1, 2, 3, 4, 7, 10, 12, 1000
, 8, 10, 800
, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 600
, 3, 5, 6, 7, 8, 800
, 1, 3, 4, 6, 8, 10, 11, 13, 1100
, 1, 3, 4, 6, 8, 9, 10, 11, 12, 13
, 1, 3, 6, 8, 11, 12, 13, 1300
, 1, 3, 4, 6, 8, 9, 11, 13, 900
, 1, 2, 3, 4, 5, 6, 7, 8, 9, 12, 13, 800
, 1, 3, 6, 7, 8, 9, 12, 13, 100
, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 400
)
# end scan(), need one blank line after last data value

# initialize type "all" or "most" with NA (not available)
type <- rep(NA, length(outcomes));
# create a data.frame df.13 with both outcomes and type
df.13 <- data.frame(outcomes, type)
# assign type to "all" if outcome is < 100
df.13$type[(df.13$outcomes < 100)] <- "all"
# assign type to "most" if outcome is >= 100
df.13$type[(df.13$outcomes >= 100)] <- "most"

```

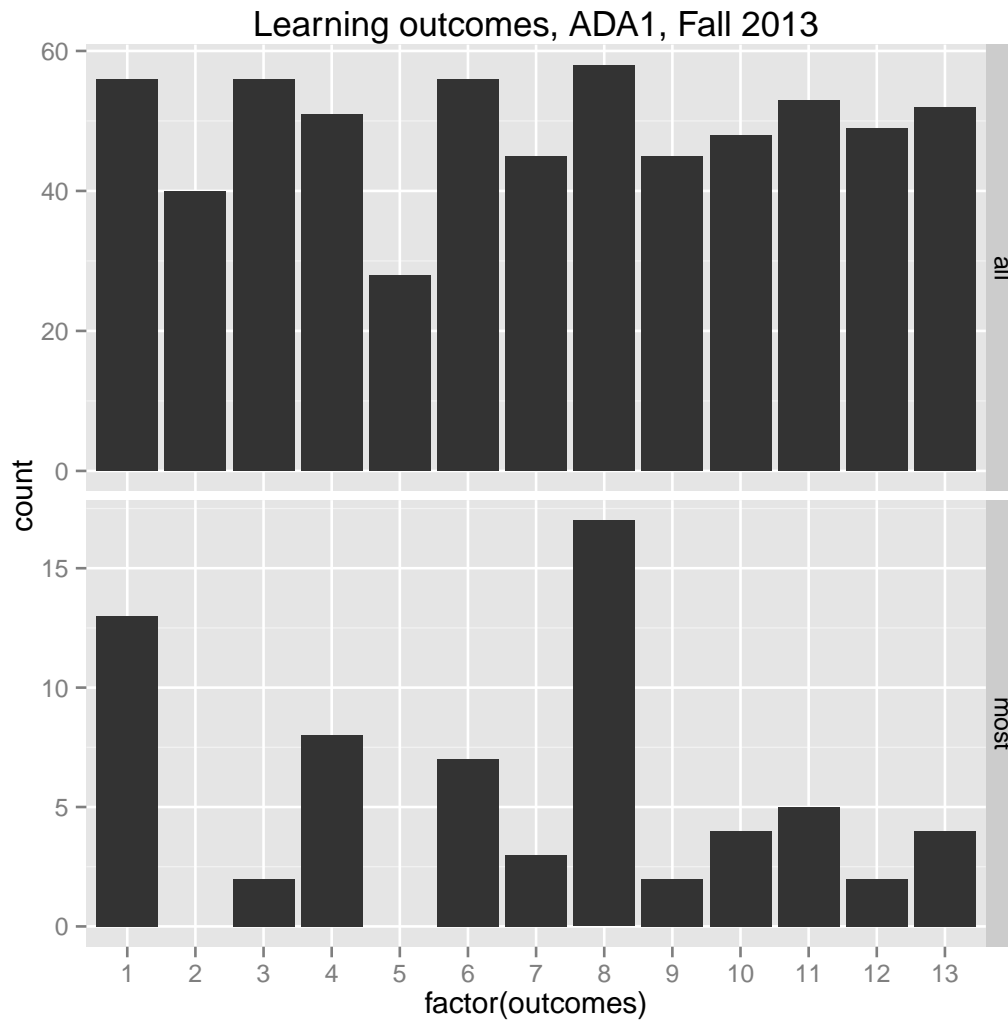
```
# divide the most outcome numbers by 100 to put on original scale
df.13$outcomes[(df.13$type == "most")] <- df.13$outcomes[(df.13$type == "most")]/100

# tabular results
xtabs( ~ outcomes + type, data = df.13)

##           type
## outcomes all most
##      1   56   13
##      2   40    0
##      3   56    2
##      4   51    8
##      5   28    0
##      6   56    7
##      7   45    3
##      8   58   17
##      9   45    2
##     10   48    4
##     11   53    5
##     12   49    2
##     13   52    4

# load the ggplot2 plotting library
library(ggplot2)
# init plot where outcomes are on the horizontal axis
p <- ggplot(df.13, aes(x = factor(outcomes)))
# add bars
p <- p + geom_bar()
# separate the plots by type "all" and "most", let vertical scale vary per plot
p <- p + facet_grid(type ~ ., scales = "free")
# Title the plot
p <- p + labs(title = "Learning outcomes, ADA1, Fall 2013")
# Print the plot
print(p)

#ggsave("C:/Dropbox/UNM/teach/ADA1_stat527/assess/ADA1_HW_00_F13_plot.png", p, height = 6, width = 8)
```



Results of Fall 2014:

```
# ADA1 Stat 427/527 Fall 2014
# 8/19/2014 4:23PM
# learning outcomes from HW00
```

```
# With so many values to enter by hand, I needed to come up with
# a quick and easy way of getting the data into R.
# I decided to type all the values from each sheet on a row,
# originally I used scan() for direct copy/paste into to R console.
# The most important outcome is multiplied 100 so it's easy to code these later.
```

```
outcomes <- c(
  1, 4, 6, 7, 8, 12, 13, 1200
, 1, 2, 3, 5, 6, 8, 12, 600
, 1, 5, 7, 9, 11, 12, 13, 1200
, 4, 9, 10, 11, 1100
, 2, 4, 5, 7, 9, 11, 13, 700
, 2, 7, 10, 1000
, 1, 3, 4, 6, 8, 11, 12, 13, 600
, 6, 7, 8, 12, 13, 1300
, 1, 3, 4, 6, 8, 10, 11, 13, 300
, 3, 4, 6, 10, 12, 300
, 2, 4, 7, 700
, 1, 2, 3, 4, 6, 8, 9, 10, 11, 12, 400
```

, 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 13, 300
, 3, 5, 6, 8, 600
, 1, 3, 6, 300
, 1, 2, 3, 6, 7, 8, 800
, 2, 4, 6, 7, 8, 10, 11, 12, 13, 1000
, 1, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 800
, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 100
, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 600
, 1, 8, 100
, 1, 2, 3, 5, 6, 7, 8, 9, 800
, 1, 3, 4, 6, 8, 9, 10, 11, 12, 13, 800
, 2, 4, 9, 12, 900
, 1, 4, 6, 8, 10, 11, 13, 800
, 1, 4, 6, 7, 11, 1100
, 2, 3, 4, 5, 6, 7, 9, 11, 12, 13, 900
, 1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 13, 1100
, 3, 4, 6, 9, 13, 900
, 1, 3, 6, 8, 12, 13, 800
, 1, 3, 6, 7, 10, 11, 12, 1100
, 1, 5, 8, 12, 800
, 1, 3, 4, 8, 10, 12, 13, 800
, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 600
, 2, 4, 6, 8, 10, 11, 12, 13, 1300
, 1, 3, 4, 6, 7, 8, 11, 13, 100
, 1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 1000
, 2, 3, 4, 6, 7, 8, 10, 12, 13, 800
, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 400
, 1, 6, 8, 9, 10, 11, 12, 13, 1100
, 1, 2, 3, 5, 8, 13, 800
, 1, 7, 9, 11, 12, 13, 900
, 3, 7, 8, 300
, 1, 3, 4, 6, 8, 11, 13, 600
, 1, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 600
, 3, 4, 6, 8, 9, 11, 12, 400
, 1, 6, 8, 12, 13, 600
, 2, 7, 9, 12, 1200
, 1, 3, 4, 6, 8, 400
, 4, 7, 8, 11, 400
, 1, 3, 6, 8, 10, 800
, 1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 600
, 1, 2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 300
, 1, 3, 6, 8, 9, 13, 1300
, 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 13, 1300
, 1, 2, 3, 4, 6, 7, 9, 12, 700
, 4, 11, 1100
, 3, 8, 11, 12, 800
, 1, 2, 3, 6, 7, 8, 9, 10, 11, 1100
, 1, 2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 1100
, 6, 8, 9, 12, 600
, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 1200
, 1, 3, 4, 6, 8, 10, 11, 12, 600
, 1, 3, 6, 7, 8, 10, 11, 800
, 3, 4, 8, 12, 300
, 2, 3, 8, 10, 11, 13, 300
, 1, 3, 4, 6, 8, 11, 12, 13, 600
, 1, 2, 3, 4, 6, 8, 11, 13, 800
, 2, 3, 4, 8, 12, 13, 800
, 1, 3, 4, 5, 6, 7, 8, 9, 11, 12, 13, 800
, 1, 3, 4, 6, 12, 13, 1200
, 1, 3, 6, 8, 11, 13, 100
, 1, 4, 5, 6, 9, 10, 11, 12, 600
, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 100
, 1, 2, 3, 5, 6, 7, 8, 9, 10, 12, 13, 800
, 1, 3, 6, 8, 9, 12, 13, 800
, 6, 8, 9, 11, 800
, 6, 8, 10, 600
, 8, 12, 800


```

, 1, 3, 4, 6, 8, 10, 11, 12, 13, 1300
, 1, 5, 7, 8, 13, 100
, 1, 2, 3, 4, 7, 8, 10, 11, 12, 1100
, 4, 6, 8, 9, 12, 600
, 1, 6, 11, 12, 100
, 1, 3, 6, 7, 8, 12, 13, 800
, 3, 4, 6, 8, 11, 13, 400
, 1, 3, 4, 6, 7, 8, 12, 1200
, 1, 3, 4, 6, 7, 8, 12, 1200
, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 800
, 1, 4, 7, 9, 10, 11, 13, 900
, 1, 2, 6, 8, 9, 12, 13, 1200
, 1, 2, 3, 4, 5, 6, 7, 9, 10, 12, 300
, 9, 10, 11, 1100
, 1, 4, 10, 400
, 1, 2, 3, 4, 6, 8, 9, 10, 11, 12, 13, 800
, 3, 6, 9, 11, 13, 600
, 1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 1300
, 3, 6, 7, 8, 9, 800
)
# end scan(), need one blank line after last data value

# initialize type "all" or "most" with NA (not available)
type <- rep(NA, length(outcomes));
# create a data.frame df.14 with both outcomes and type
df.14 <- data.frame(outcomes, type)
# assign type to "all" if outcome is < 100
df.14$type[(df.14$outcomes < 100)] <- "all"
# assign type to "most" if outcome is >= 100
df.14$type[(df.14$outcomes >= 100)] <- "most"
# divide the most outcome numbers by 100 to put on original scale
df.14$outcomes[(df.14$type == "most")] <- df.14$outcomes[(df.14$type == "most")]/100

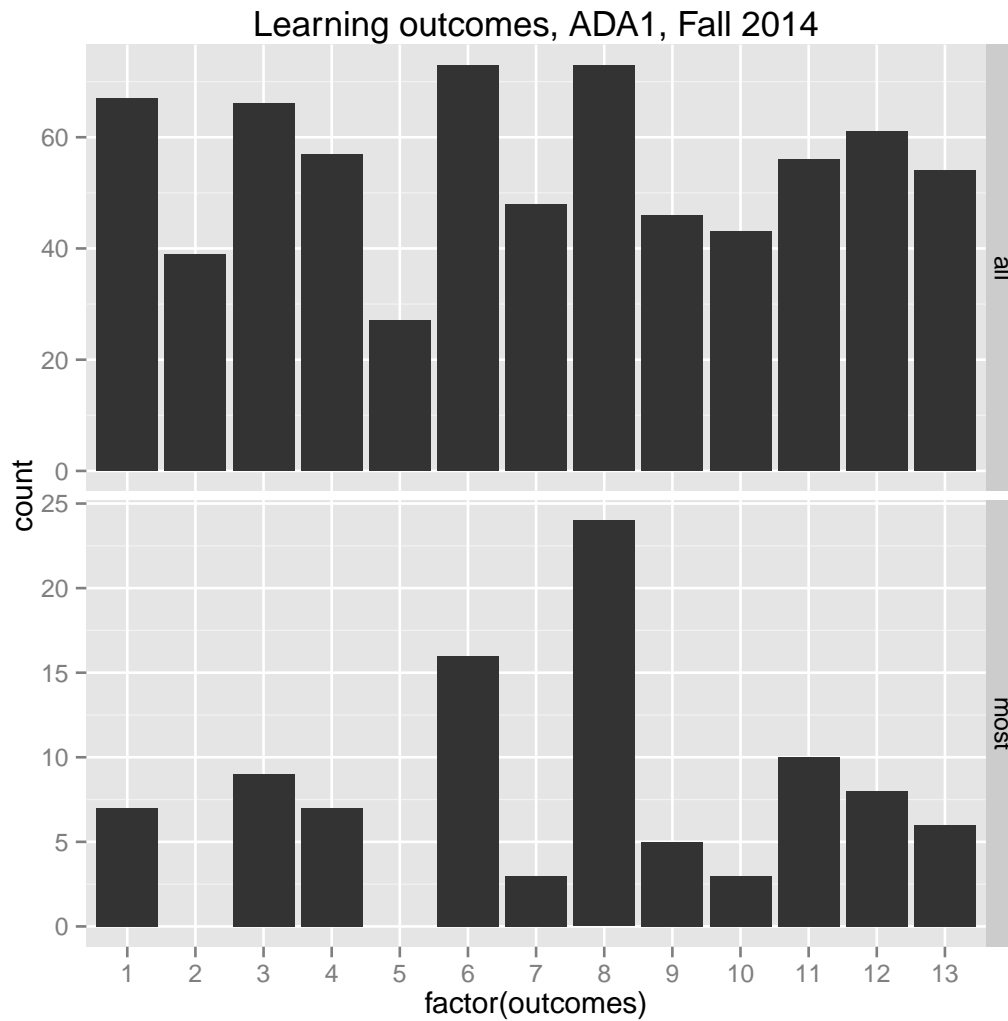
# tabular results
xtabs( ~ outcomes + type, data = df.14)

##           type
## outcomes all most
##      1     67    7
##      2     39    0
##      3     66    9
##      4     57    7
##      5     27    0
##      6     73   16
##      7     48    3
##      8     73   24
##      9     46    5
##     10     43    3
##     11     56   10
##     12     61    8
##     13     54    6

# load the ggplot2 plotting library
library(ggplot2)
# init plot where outcomes are on the horizontal axis
p <- ggplot(df.14, aes(x = factor(outcomes)))
# add bars
p <- p + geom_bar()
# separate the plots by type "all" and "most", let vertical scale vary per plot
p <- p + facet_grid(type ~ ., scales = "free")
# Title the plot
p <- p + labs(title = "Learning outcomes, ADA1, Fall 2014")
# Print the plot
print(p)

#ggsave("C:/Dropbox/UNM/teach/ADA1_stat527/assess/ADA1_HW_00_F14_plot.png", p, height = 6, width = 8)

```



Combined results:

```
# ADA1 Stat 427/527 Combined 2012 - current

# create a year indicator column
df.12$year <- 2012
df.13$year <- 2013
df.14$year <- 2014
# combine datasets together
df.all <- rbind(df.12, df.13, df.14)
df.all$year <- factor(df.all$year)

# tabular results
xtabs( ~ outcomes + type, data = df.all)

##      type
## outcomes all most
##      1  181  28
##      2  107   1
##      3  177  20
##      4  145  18
##      5   72   0
##      6  190  39
##      7  141   8
##      8  206  61
```

```
##      9 122  11
##     10 132  10
##     11 160  24
##     12 152  13
##     13 157  19

# load the ggplot2 plotting library
library(ggplot2)
# init plot where outcomes are on the horizontal axis
p <- ggplot(df.all, aes(x = factor(outcomes)))
# add bars
p <- p + geom_bar(aes(fill = year), colour = "black")
# separate the plots by type "all" and "most", let vertical scale vary per plot
p <- p + facet_grid(type ~ ., scales = "free")
# Title the plot
p <- p + labs(title = "Learning outcomes, ADA1, Fall 2014")
# Print the plot
print(p)

#ggsave("C:/Dropbox/UNM/teach/ADA1_stat527/assess/ADA1_HW_00_all_plot.png", p, height = 6, width = 8)
```

