HW 2

Adon Rosen

9/29/2019

Load the library(s)

```
library(foreign) ## Will be used to load .sav file
library(ggplot2) ## Will be used for plotting
library(grid) ## More plotting functions
library(knitr) ## For cosmetics
print("Done loading librarys")
```

```
## [1] "Done loading librarys"
```

Declare some knitr stuff

```
hook_output = knit_hooks$get('output')
knit_hooks$set(output = function(x, options) {
  # this hook is used only when the linewidth option is not NULL
  if (!is.null(n <- options$linewidth)) {
    x = knitr:::split_lines(x)
    # any lines wider than n should be wrapped
    if (any(nchar(x) > n)) x = strwrap(x, width = n)
    x = paste(x, collapse = '\n')
  }
  hook_output(x, options)
})
```

Declare a function we will use later

```
summarySE <- function(data=NULL, measurevar, groupvars=NULL, na.rm=FALSE,
                      conf.interval=.95, .drop=TRUE) {
    library(plyr)

    # New version of length which can handle NA's: if na.rm==T, don't count them
    length2 <- function (x, na.rm=FALSE) {
        if (na.rm) sum(!is.na(x))
        else       length(x)
    }

    # This does the summary. For each group's data frame, return a vector with
    # N, mean, and sd
    datac <- ddply(data, groupvars, .drop=.drop,
      .fun = function(xx, col) {
        c(N    = length2(xx[[col]], na.rm=na.rm),
          mean = mean   (xx[[col]], na.rm=na.rm),
          sd   = sd     (xx[[col]], na.rm=na.rm)
```

```
        )
      },
      measurevar
    )

    # Rename the "mean" column
    datac <- rename(datac, c("mean" = measurevar))

    datac$se <- datac$sd / sqrt(datac$N)  # Calculate standard error of the mean

    # Confidence interval multiplier for standard error
    # Calculate t-statistic for confidence interval:
    # e.g., if conf.interval is .95, use .975 (above/below), and use df=N-1
    ciMult <- qt(conf.interval/2 + .5, datac$N-1)
    datac$ci <- datac$se * ciMult

    return(datac)
}
```

Answer Question 1 here

Question 1.A

```
## [1] "We are told in the problem that the mean of the
sample of OU undergrads is 25"
```

```
## [1] "We will use this value as our point estimate of
mean depression"
```

Question 1.B

```
## [1] "The margin of error is: 2.03010792825034"
```

Question 1.C

```
## [1] "The 95% confidence interval for mean OU undergrad
depression is: [22.97,27.03]"
```

Question 1.D

```
## [1] "The 95% confidence interval of the mean OU
undergrad depression rating reflects the variation of mean
values we would obtain if the sampiling procedure was
repated a large number of times."
```

```
## [1] "Explicitly, we would expect 95% of the means to
fall within 22.97 and 27.03 from this resampling procedure"
```

Question 1.E

```
## [1] "Given the fact that the national undergraduate mean
depression is estimated at 40, and the upper limit of the
OU undergraduate mean depression confidence interval was
27.03 we can confidentley state that these two populations
exhibit different means"
```

Answer question 2 here

```
## [1] "A p-value reflects the probability of observing an
effect of equal or greater absolute magnitude given a true
null hypothesis."
```

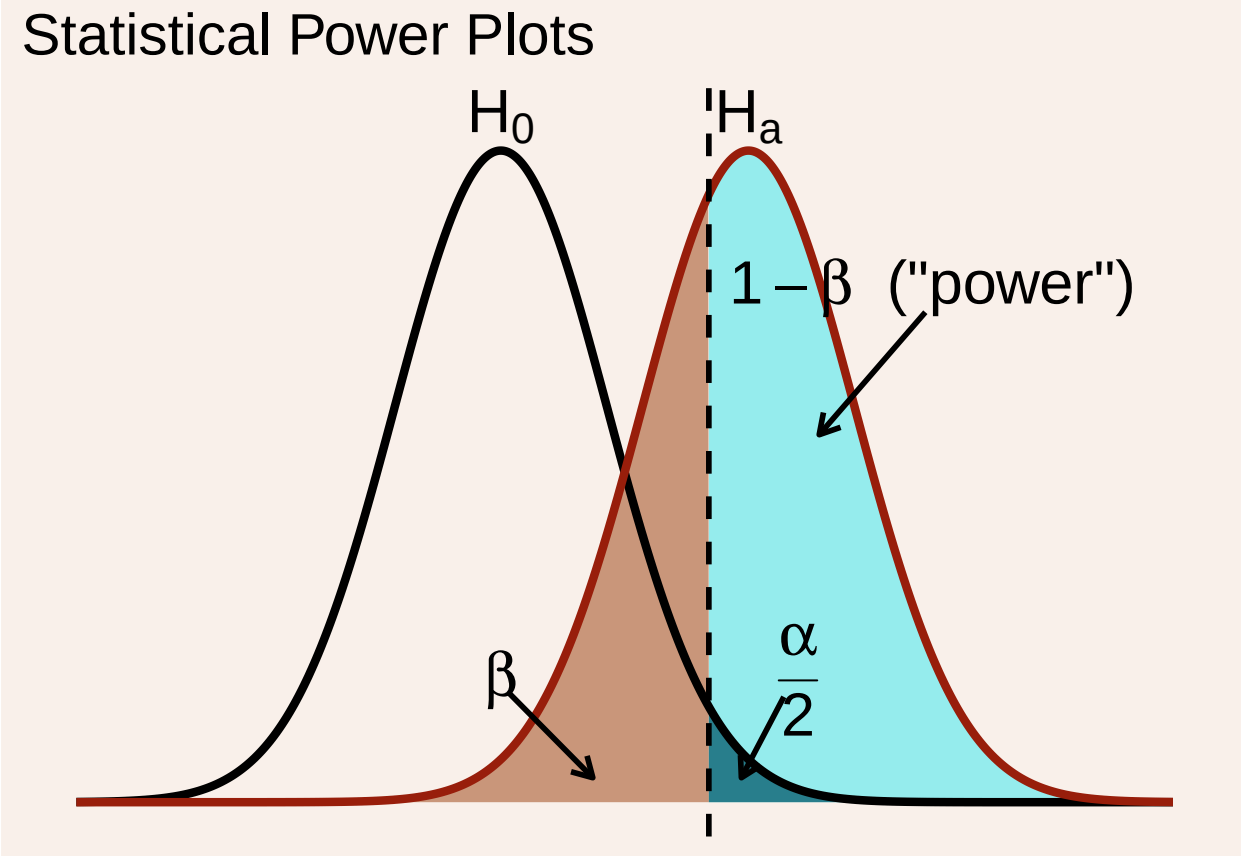
Answer question 3 here

```
## Warning: `show_guide` has been deprecated. Please use `show.legend`
## instead.

## Warning: `show_guide` has been deprecated. Please use `show.legend`
## instead.

## Warning: `show_guide` has been deprecated. Please use `show.legend`
## instead.

## Warning in is.na(x): is.na() applied to non-(list or vector) of type
## 'expression'
```



Answer question 4 here

```
## [1] "Statistical power is influenced by:"
## [1] "1. Sample size"
## [1] "2. Estimated effect sizes"
## [1] "3. One vs Two directional tests"
```

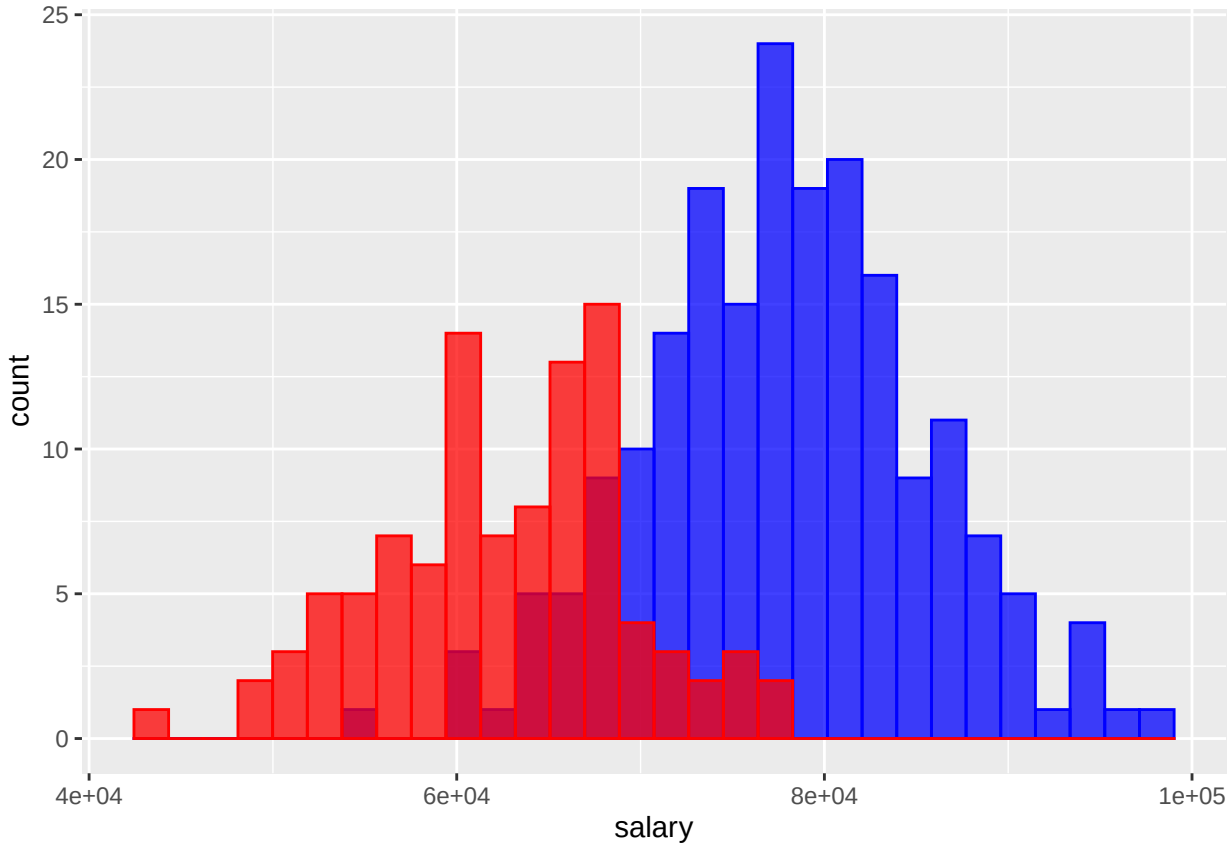
Answer question 5 here

```
## First load the data
in.data <- read.csv("../HW1/salary.csv")
```

Question 5.A

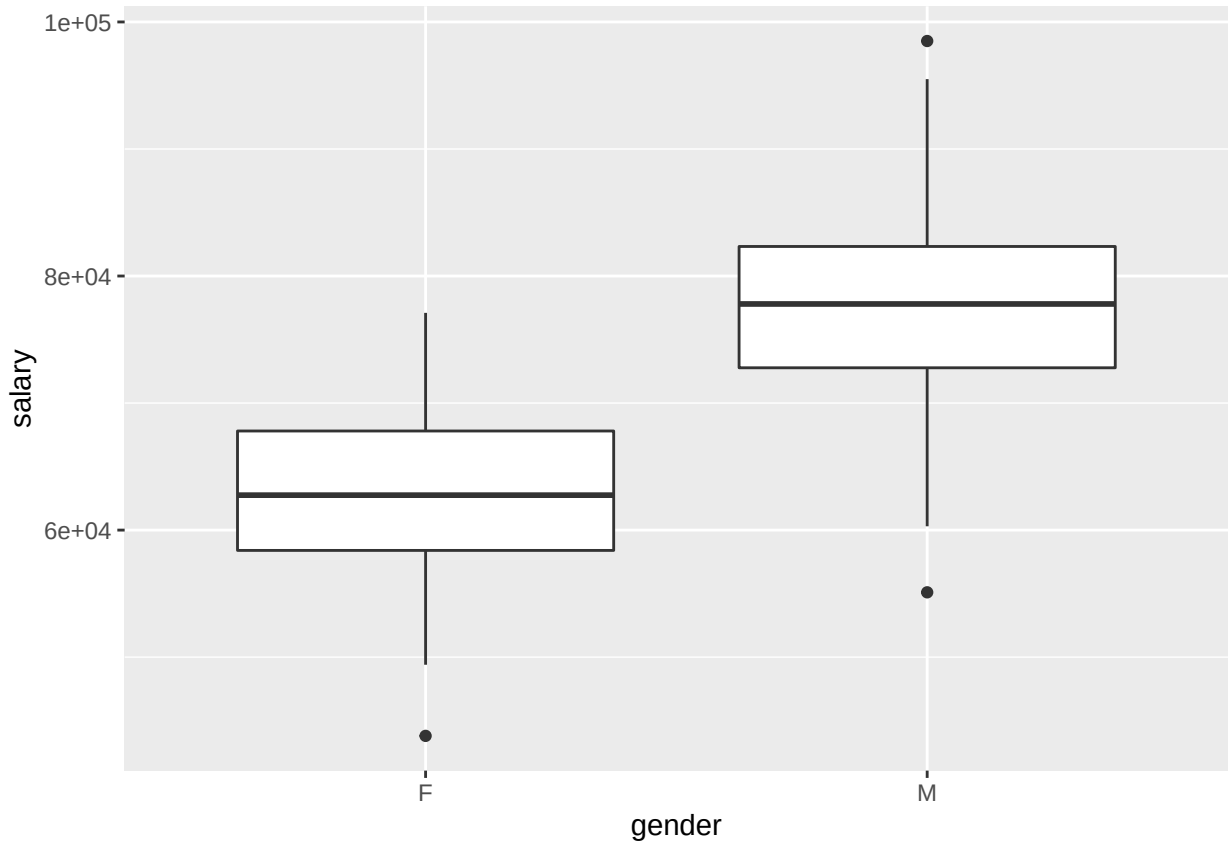
```
ggplot(in.data) +
  geom_histogram(data=in.data[which(in.data$gender=="M"),],aes(x=salary), color='blue', alpha=.75, fill=
  geom_histogram(data=in.data[which(in.data$gender=="F"),],aes(x=salary), color="red", alpha=.75, fill=
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



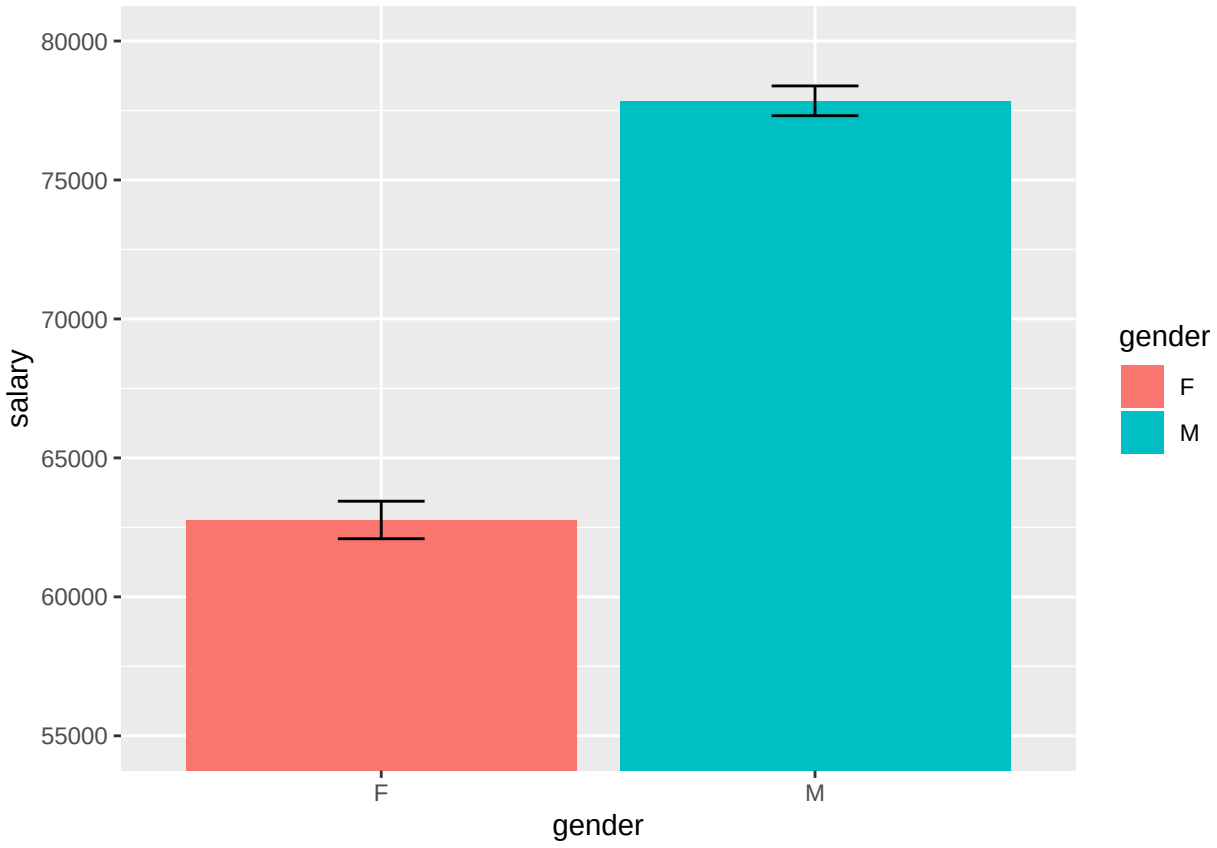
Question 5.B

```
ggplot(in.data, aes(x=gender, y=salary)) +
  geom_boxplot()
```



Question 5.C

```
plot.data <- summarySE(data=in.data, measurevar = 'salary', groupvars = 'gender')
ggplot(plot.data, aes(x=gender, y=salary, fill=gender)) +
    geom_bar(position=position_dodge(), stat="identity") +
    geom_errorbar(aes(ymin=salary-se, ymax=salary+se),
                  width=.2,
                  position=position_dodge(.9)) +
    coord_cartesian(ylim=c(55000, 80000))
```


80000
75000
70000
65000
60000
55000
salary
F
M
gender
gender
F
M