Homework 1

INF 511

Muhammad

1 Some R Basics

1.1 `faraway` loading package (1-point)

```
# Load the `faraway` package
#if (system.file(package='faraway') != TRUE){
#  install.packages('faraway')
#}

library("faraway")
```

1.2 Working with `data.frame` objects

```
toy_df <- data.frame(
    name=c("Fred", "Ethyl", "Ricky", "Lucy","Babalu"),
    program=c("surfing", "surfing", "singing","singing", "dancing"),
    gpa=c(3.2, 3.4, 3.4, 3.3, 4.0),
    sat=c(1200, -999, 1300, 1250, 1600))
```

1.2.1 Summarize (1 point)

Creating Summary of the data frame

```
summary(toy_df)
```

```
     name              program               gpa              sat
 Length:5           Length:5           Min.   :3.20   Min.   :-999.0
 Class :character   Class :character   1st Qu.:3.30   1st Qu.:1200.0
 Mode  :character   Mode  :character   Median :3.40   Median :1250.0
                                       Mean   :3.46   Mean   : 870.2
                                       3rd Qu.:3.40   3rd Qu.:1300.0
                                       Max.   :4.00   Max.   :1600.0
```

1.2.2 Subsetting (2 points)

Used the `$` notation to subset the `gpa` column of the `toy_df` object, and used the `mean()` function to calculate the average of the column.

```r
gpa <- toy_df$gpa
mean(gpa)
```

```
[1] 3.46
```

1.2.3 Levels (4 points)

Used the $ notation again to subset the `program` column of the `toy_df` object. Convert the `program` column, which is currently a `character` string, to a `factor` variable. Then, use the `levels()` function to print the levels of this new factor object.

```r
program <- toy_df$program
y <- factor(program)
levels(y)
```

```
[1] "dancing" "singing" "surfing"
```

1.3 Vectorized functions

Use the following vector to complete the sub-tasks below:

```r
my_vec = seq(from=1,to=5,by=0.8)
```

1.3.1 Calculate the length of `my_vec`. (1 point)

Used a built-in function within R to output the length of `my_vec`.

```r
length(my_vec)
```

```
[1] 6
```

1.3.2 Calculate the natural log of each element of `my_vec`. (1 point)

Used a built-in function within R to output the natural log of each element of `my_vec`. It is a single function on a single line of code

```r
log(my_vec)
```

```
[1] 0.0000000 0.5877867 0.9555114 1.2237754 1.4350845 1.6094379
```

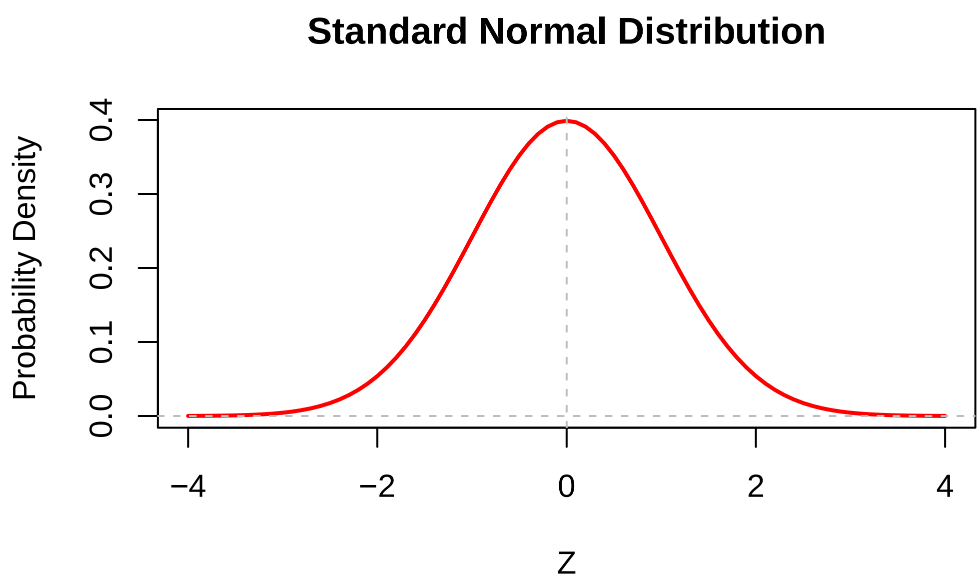
2 Probability distributions

2.1 Standard normal (5 points)

Plot the probability distribution function (as a curve) that describes the "standard normal," which is the normal distribution with mean zero and standard deviation equal to one. In other words, plot $P(z|\mu = 0, \sigma = 1)$ for a range of continuous random variable z, where $z \sim N(\mu = 0, \sigma^2 = 1)$. Make sure that z ranges from -4 to 4. Label the axes appropriately.

```r
mu <- 0
sigma <- 1
x <- seq(-4, 4, by=0.1)
```

```
pdf <- 1 / (sqrt(2 * pi * sigma^2)) * exp(-((x - mu)^2) / (2 * sigma^2))
plot(x, pdf, xlab = "Z", ylab = "Probability Density", main = "Standard Normal Distribution", type = "l"
abline(h = 0, col = "gray", lty = 2)
abline(v = 0, col = "gray", lty = 2)
```



2.2 CDF (2 points)

Used R to calculate $P(z \leq 1.645 | \mu = 0, \sigma = 1)$

```
pnorm(1.645, mean=0, sd=1)
```

```
[1] 0.9500151
```

2.3 Inverse CDF (2 points)

Used the `qnorm()` to calculate the value of z that delineates that 95% of the standard normal probability distribution falls below this value of z. This demonstrates the inverse CDF. You should see a relationship with the answer of the above question.

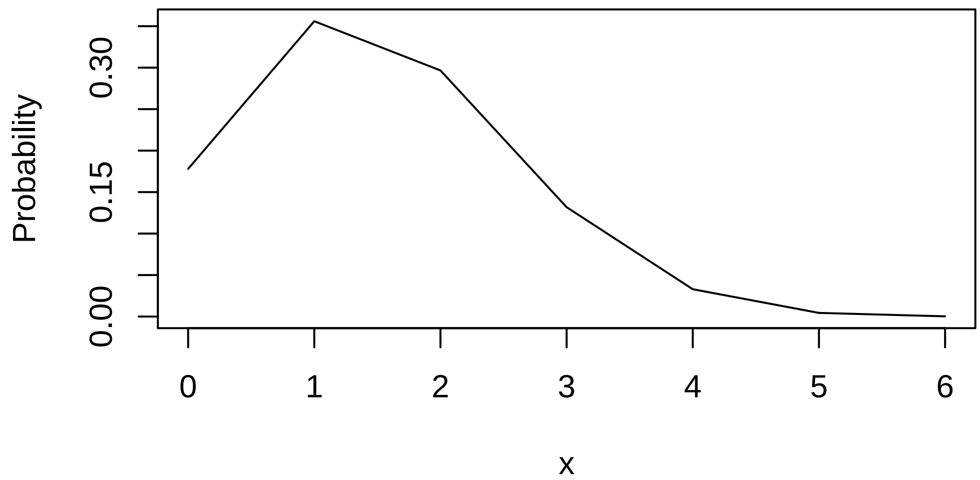
```
qnorm(0.95, mean=0, sd=1)
```

```
[1] 1.644854
```

2.4 Binomial distribution (5 points)

Used R to plot the binomial probability mass function with $n = 6$ (i.e., `size`=6) and $p = .25$. Because the binomial is a discrete probability distribution, this plot formatted similarly to the Poisson example

```r
n <- 6
p <- 0.25
x <- 0:n
pmf <- dbinom(x, size=n, prob=p)
plot(x, pmf, type="l", xlab="x", ylab="Probability")
```



2.5 CDF of Binomial (2 points)

Used R to compute $P(Y \geq 2)$ when $Y \sim binomial(n = 6, p = 0.25)$. Be careful with the sign of the inequality.

```r
n <- 6
p <- 0.25
y <- 2

1 - pbinom(y-1, size=n, prob=p)
```

```
[1] 0.4660645
```

3 Algebraic expressions

Consider Y_1 and Y_2, which are *independent* random variables with means (i.e., *expectations*) equal to μ_1 and μ_2, respectively, and variances σ_1^2 and σ_2^2, respectively.

3.1 What is the mean of the linear expression? (2 points)

$mean\_2y1\_5\_8y2 \leftarrow (2 \cdot \mu 1) + 5 + (8 \cdot \mu 2)$

```r
mu1 <- 3
mu2 <- 4
```

```r
mean_2y1_5_8y2 <- (2 * mu1) + 5 + (8 * mu2)
mean_2y1_5_8y2
```

```
[1] 43
```

3.2 What is the variance? (2 points)

$total_var \leftarrow 4 \cdot y1_var + 64 \cdot y2_var$

```r
y1_mean <- 3
y2_mean <- 4
y1_var <- 2
y2_var <- 3
total_var <- 4 * y1_var + 64 * y2_var
total_var
```

```
[1] 200
```

3.3 What is the distribution? (2 points)

If Y_1 and Y_2 are both normally distributed, what is the distribution of the linear combination $2Y_1+5+8Y_2$? Moreover, what are the parameters that describe this distribution?

```r
# Generate Y1 and Y2
n <- 1000 # number of observations
mu1 <- 0 # mean of Y1
sigma1 <- 1 # standard deviation of Y1
mu2 <- 5 # mean of Y2
sigma2 <- 2 # standard deviation of Y2

Y1 <- rnorm(n, mu1, sigma1)
Y2 <- rnorm(n, mu2, sigma2)

# Calculate the mean and variance of the linear combination
mean <- 2 * mu1 + 5 + 8 * mu2
var <- 4 * sigma1^2 + 64 * sigma2^2
mean
```

```
[1] 45
```

```r
var
```

```
[1] 260
```

3.4 What is the covariance? (1 point)

What is the covariance between Y_1 and Y_2?

```r
cov <- cov(Y1, Y2)
cov
```

```
[1] 0.06669114
```