R Studio

TIP: There is an object and an entity of each q e.g

Create two histograms one above the other for chick weights using R. The first histogram should be for chick weights fed “sunflower” and the second for those fed “horsebean”
OBJECT = Weight

Entitiy = Feed Type

To find the product of something

```
Prod(26:21)
```

To find the mean of something

```
simpson_mean <- mean(imdb_rating, na.rm = T)
simpson_mean
```

To find the standard deviation of something

```
ssd <- sd(imdb_rating, na.rm = T)
```

To plot a histogram

```
hist(imdb_rating, breaks=c(0:10), main='Distribution of IMDB Ratings for Simpsons')
```

To plot a histogram from a particular time period

```
par(mfrow=c(2,1))

hist(imdb_rating[season > 1 & season <= 10], breaks=c(5:10), main='Comparing distribution of ratings from Seasons 1-10 (top) and Season 11-
hist(imdb_rating[season > 10 & season <= 20], breaks=c(5:10), main='', xlab='IMDB Rating')
```

To do a boxplot

```
boxplot(imdb_rating, ylab='IMDB Rating', main='Distribution of Simpsons Episode Ratings')
```

To do a box plot including a y lim

```
boxplot(imdb_rating[season==10], ylim=c(6.5,8.5), ylab='IMDB Rating', main='Distribution of ratings for Season 10')
```

To do a box plot showing average

```
plot( tapply(imdb_rating, season, mean, na.rm=T), xlab='Season number', ylab='IMDB Rating', main='Simpsons IMBD Rating over time', type='b'
```

To change colour type

```
type='o', col='green'
```

To make a part of a box plot a certain colour

```
plot(tapply(imdb_rating[number_in_season >= 1 & number_in_season <10], season[number_in_season >= 1 & number_in_season <10], mean, na.rm=T)
, xlab='Season number', ylab='IMDB Rating', main='Simpsons IMBD Rating over time', type='o', col='green')
```

To do a scatter plot

```
plot( number_in_season[season == 2], us_viewers_in_millions[season == 2], xlab='Episode number in Season 2', ylab='US viewers (in millions)
```

To draw best fit line

```
abline(lm(us_viewers_in_millions[season == 2] ~ number_in_season[season == 2]))
```

To see relationship in a box plot

```
boxplot(imdb_rating ~ season)
```

To see highest

```
which.max( tapply(imdb_rating, season, mean, rm.na=T) )
```

Law of Diminishing Returns Graph

```
k<- 1:30 # simulating 1 to 30 additional components
p1 <- .5 # reliability of a single component
diffs_p = 0 # differences in probabilities for increasing components
probs <- 1-(1-p1)^k # calculating probabilities of each additional component (for k from 1 to 30)
for (i in 2:30) diffs_p[i-1] <- probs[i] - probs[i-1]
plot(diffs_p, type='o', main='Law of diminishing returns', xlab='No of Additional (Parallel) Components', ylab='Gain in reliability')
```

To plot a regular histogram

```
par(mfrow=c(2,1))
hist(weight[feed == 'sunflower'], main='Comparing food types ', xlab='Food type')
```

To Plot a regular boxsplot

```
par(mfrow=c(1,1))
boxplot(weight ~ feed, col="orange", main="Distribution of Feed types/Weight", ylab="Weight in Grams", xlab="Feed Type")
```

To find mean of non integers

```
tapply(weight, feed, mean)
```

To find the variance of none integers

```
tapply(weight, feed, var)
```

To Make side by side boxplot

```
boxplot(imdb_rating[5<=season & season<10], imdb_rating[season>10 & season<15], names=c("Season 5-10", "Season 10-15"),
main='Comparing seasons of Simpsons by IMDB Rating',
ylab='IMDB Rating' )
```

To find proportions

```
sum(imdb_rating <= 6, na.rm =T) / length(imdb_rating [!is.na(imdb_rating)])
```

To find proportions between two numbers

```
sum(7 < imdb_rating & imdb_rating < 8, na.rm =T) / length(imdb_rating [!is.na(imdb_rating)])
```

To create 2 histograms or just do it one by one with the above

```
par(mfrow=c(2,1))
hist(weight[feed=='sunflower'], breaks=seq(0,500, 50), main='Chick weights for sunflower (top) vs horsebean (bottom)', xlab='') hist(weight
```

Sample Q

What proportion of chicks weigh less 159 grams that have been fed linseed or sunflower. **Show your R code.**

```
n_weights <- weight[feed=='linseed' | feed=='sunflower'] proportion <- length(n_weights[n_weights < 159]) / length(weight) 2.816901% of the
```

Probability of getting something

```
sum(cards$suit == "Heart") / nrow(cards)
```

- To get two things together use an "and" statement. To get the "or" add them together then minus the combination of both

To create a subset of data

```
e.g. heart <- subset(cards, cards$suit=="Heart")
```