*All Assignment files @ turmang/BIOL497R/Assignment_Files/Assgn1 from primary dir

Q1.

A) Based on visual inspection (Please note spaces included for readability, not literal spaces in file context). It's looks like a tsv with various feature information in each of the 9 columns.
Chr# \t organismInfo(?) \t featureType \t StartBasePairLoc \t EndBasePairLoc \t spacerMaybe \t strandInfo \t moreSpacer \t AccessionID/Name
Code Used: head Egrandis_297_v2.0.gene_exons.gff3

B) 253249
Code Used: grep -c "exon" Egrandis_297_v2.0.gene_exons.gff3

C) In Assgn1 files
Code Used: grep "gene" Egrandis_297_v2.0.gene_exons.gff3 > Egrandis_297_v2.0.geneonlyGT.gff3

D) 36349
Code Used: grep -c "gene" Egrandis_297_v2.0.gene_exons.gff3

Q2.

A) In Assgn1 files
Code Used: (1) awk '{print $9,$1,$7,$4,$5}' Egrandis_297_v2.0.geneonlyGT.gff3 > Egrandis_genelocationsGT.txt
(2) sed -i 's/ID=\S*;Name=//g' Egrandis_genelocationsGT.txt

B) 18244, mean=3104 bp, longest=56414 bp, shortest=200 bp
Code Used: (1) grep -c "+" Egrandis_genelocationsGT.txt
(2) bash meanShortLong.sh
(3) awk '{s+=$1} END {print s}' geneLens.txt
(4) wc -l geneLens.txt //3&4 validation steps

C) In Assgn1 files
Code Used: (1) awk '{print $5-$4}' Egrandis_genelocationsGT.txt > geneLens.txt
(2) Rscript doHist.R //Note this doesn't seem to run properly on Jupyter's terminal, but it runs fine for me – produces PDF with something stored in it, but is view-wise 'blank' (In Jupyter). File is in Assgn1 files for viewing.

Q3.

A) In Assgn1 files

Q4.

A) 63557Kb and(?)2cd7421f6bdf4a687a5e5550a233b16d
Code Used: (1) ls –l //for size

(2) md5sum Illumina_50bp.fastq

B) In Assgn1 files
Code Used: grep ":ATCACG" Illumina_50bp.fastq > identifiers_combined.txt

Q5.

A) one
B) end of line
C) Not – e.g. [^c] don't match the literal char c
D) abcb

Q6. A) Used script grepStuff.R, in Assgn1 files

grep() and gregexpr() both use regular expressions to search for pattern matches in a designated character string, but they have different flavors of returning indicies, returning 0 matches, and match evaluation. grep() returns the indices of the character vector containing a match while gregexpr() returns the indices of the actual string where the matches reside. So for each individual gene line containing the searched for sequence, grep() returns 1 for the index because it finds a match in the first character vector which is the entire line, while gregexpr() returns the actual points in the String where the matches are found. Additionally using gregexpr() you can get the length of indexed matches. When grep() doesn't find any matches, the return from the actual function is 0, but for gregexpr() the return is instead 1 with and **index** of -1. In the script I've written I handle these outputs with checks to make it more clear what the interpreted return is (as opposed to having the print statement gregexpr() found 1 match with an index of -1).