Data Manipulation with Grep and Humdrum Tools:

Basic Humdrum Syntax: Output to file

<appname> source-file > target-file
<appname> -v "2 3 2" source-file > target-file

wc
---
wc [-clmw] <filename>
- returns count of lines, words, characters (if no flags used)
-c bytes
-l lines
-m characters
-w words

grep

| or STDIN

use this to send output from one program to another instead of sending to a file

e.g. "ls | grep blah"

outputs a list of all files in the current directory with "blah" in their namej

. single character wildcard
* multi-character wildcard

Regular Expressions - wildcards on steriods

Things to remember:

escape characters -> "\"

e.g. to search for hello.gif in our data the "." will be incorrectly seen as a single-character wildcard.

solution: grep 'hello\.gif' file

where the \. refers to an "escaped" character

recursive grep/find:

find . -exec grep -l "text-to-be-found" {} \; > outputfile.txt

- searches all sub-directories starting at "." For string and outputs filenames to file

Grep meta-characters		
Function	**Character**	**Example**
Match zero or more instances	*	`grep 'sonata*'` would match sonata and sonatas
Match from beginning of line (record) only	^	`grep '^!!!'` would match all comment records in a Humdrum file
Match from end of line only	$	`grep '$-'` would match the rightmost end-of-spine marker [*-] in a Humdrum file
Match any single character	.	`grep 'p.n'` would match pan or pin
Turn off the meta-function of the next character	\	`grep '\$11\.00'` would match $11.00
		`grep '^\*\*kern'` would match interpretation records beginning **kern

Some decent tutorials:

http://pegasus.rutgers.edu/~elflord/unix/grep.html
http://www.macworld.com/2004/12/secrets/jangeekfactor/index.php
http://www.itworld.com/Comp/2378/swol-1199-unix101/

Commands useful to dealing with Unix directories:

1a. Match the expression in single quotes; place matches in new file:

```
grep -h '!!!OTL:' *.krn > newfile
```

```
!!!OTL: Make New Friends
!!!OTL: Cherries So Ripe
!!!OTL: The Bell Doth Toll
!!!OTL: Orleans, Beaugency
!!!OTL: Oranges and Lemons
!!!OTL: Oh, How Lovely is the Evening
!!!OTL: Himmel und Erde
!!!OTL: Shalom, Chaverim
…
```

1b. Match the expression; sort results; place in new file:

```
grep -h '!!!OTL:' *.krn | sort > newfile
```

```
!!!OTL: Cherries So Ripe
!!!OTL: Endendez-vous?
!!!OTL: Es toenen die Lieder
!!!OTL: Frere Jacques
!!!OTL: Hey, Ho, Nobody Home
!!!OTL: Himmel und Erde
!!!OTL: Make New Friends
!!!OTL: Oh, How Lovely is the Evening
!!!OTL: Oranges and Lemons
…
```

2. Match the expression; preface matching lines with source-file name

```
grep '!!!OTL:' *.krn > newfile
```

```
round01.krn:!!!OTL: Make New Friends
round02.krn:!!!OTL: Cherries So Ripe
round03.krn:!!!OTL: The Bell Doth Toll
round04.krn:!!!OTL: Orleans, Beaugency
round05.krn:!!!OTL: Oranges and Lemons
round06.krn:!!!OTL: Oh, How Lovely is the Evening
round07.krn:!!!OTL: Himmel und Erde
round08.krn:!!!OTL: Shalom, Chaverim
round09.krn:!!!OTL: Hey, Ho, Nobody Home
round10.krn:!!!OTL: Es toenen die Lieder
round11.krn:!!!OTL: White Coral Bells
…
```