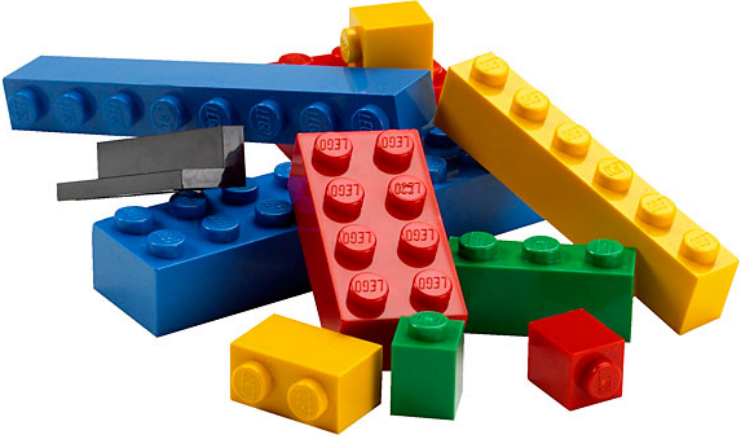
Regular Expressions

(special thanks to Mark Hansen)

Biostatistics 140.776

Regular expressions

- Regular expressions can be thought of as a combination of literals and *metacharacters*
- To draw an analogy with natural language, think of literal text forming the words of this language, and the metacharacters defining its grammar
- Regular expressions have a rich set of metacharacters



Literals

Simplest pattern consists only of literals. The literal “nuclear” would match to the following lines:

```
Why do we trust our government with 8,500 nuclear weapons,
but not to administer health care?

Beginning to regret that nuclear taco

Apple is all about nuclear family level network effects.

"I’m going to destroy Android, because it’s a stolen
product. I’m willing to go thermonuclear war on
this." - Steve Jobs
```

Literals

The literal “Obama” would match to the following lines

```
I am going to take so many pictures of Michelle Obama today...

It’s Official...The USA Has Been Renamed. It is Now the USSR (United States Socialist Republic)
Thanks Mr Obama.

I’m #Obama all the way I think 4 years is too short to be in office

Obama’s budget was rejected by the Senate 99 - 0.
```

Regular Expressions

- Simplest pattern consists only of literals; a match occurs if the sequence of literals occurs anywhere in the text being tested
- What if we only want the word "Obama", but no hashtags? Or sentences that end in the word "Clinton", or "clinton"?

Regular Expressions

We need a way to express

- whitespace word boundaries
- sets of literals
- the beginning and end of a line
- alternatives ("war" or "peace")

Metacharacters to the rescue!

Metacharacters

Some metacharacters represent the start of a line

```
^i think
```

will match the lines

```
i think should text me.

i think i lost my laptop...

i think "hipster shorts" are the ugliest stupidest
things ever

i think i'm gonna go for a run after work
today.....i think....=\
```

Metacharacters

$ represents the end of a line

```
morning$
```

will match the lines

```
i hate parting ways with my blankie in the morning

#oomf is cracking me up this morning

so watsup this morning

No plans tonight have to be to work to early in the morning

Nothin like sprinting after a bus on a Tuesday morning
```

Character Classes with []

We can list a set of characters we will accept at a given point in the match

```
[Bb][Uu][Ss][Hh]
```

will match the lines

```
Headed to King George to be with Pern, & Bushel of
whole crabs waiting and doing some party planning!

Eco design pioneer Barbara Dornbush speaking
June 17 11:30AM at Verde Home in Westside.

Kanye's as stupid as Bush.

is bush league ..... thanks for nothing!
```

Character Classes with []

```
^[Ii] am
```

will match

```
I am BEYOND irritated right now

i am boycotting the apple store

I am twittering from iPhone

i am going to dream in XML tonight. ugh.

I am so over this. I need food. Mmmm bacon...
```

Character Classes with []

Similarly, you can specify a range of letters [a-z] or [a-zA-Z]; notice that the order doesn't matter

```
^[0-9][a-zA-Z]
```

will match the lines

```
7th inning stretch
2nd half soon to begin. OSU did just win something
3am - cant sleep - too hot still.. :(
5ft 7 sent from heaven
1st sign of starvagtion
```

Character Classes with []

When used at the beginning of a character class, the “^” is also a metacharacter and indicates matching characters NOT in the indicated class

```
[^?.]$
```

will match the lines

```
i like basketballs
6 and 9
dont worry... we all die anyway!
Not in Baghdad
helicopter under water? hmmm
```

More Metacharacters

"." is used to refer to any character. So

```
9.11
```

will match the lines

```
With the anniversary of 9/11 coming up, I just
wanted to wish the world some Peace.

tune in tomorrow morning(9-11)to The Weekend Sports
Buzz on 1450am

why do i ALWAYS look at the clock at 1:43 and 9:11?????

Stock up now for the holidays at Siesel's Meats
in San Diego, CA 92110
```

More Metacharacters: |

The | character indicates a logical “or”; we can use it to combine two expressions, the subexpressions being called alternatives

```
flood|fire
```

will match the lines

```
eating some firehouse subs. m m

On way home drove by flooding areas of the San Diego
River, was okay then, but how long will it be okay
with the rain still coming down?

Between you and me, there are wildfires.
```

More Metacharacters: |

We can include any number of alternatives...

```
flood|earthquake|hurricane|coldfire
```

will match the lines

```
Not a whole lot of hurricanes in the Arctic.

We do have earthquakes nearly every day somewhere
in our State

hurricanes swirl in the other direction

coldfire is STRAIGHT!

'cause we keep getting earthquakes
```

More Metacharacters: |

The alternatives can be real expressions and not just literals

```
^[Gg]ood|[Bb]ad
```

will match the lines

```
Good morning tweeps

goodnight handsome <3

this is a bad idea for Netflix

Not a bad way to put it. Content has to be useful
to work, and must serve our readers and users.
```

More Metacharacters: (and)

Subexpressions are often contained in parentheses to constrain the alternatives

```
^([Gg]ood|[Bb]ad)
```

will match the lines

```
good for you!

Good. How is the house

Bad dreams are overrated.

Badger Herald uses really weak reasoning in Op-Ed
```

More Metacharacters: ?

The question mark indicates that the indicated expression is optional

```
[Gg]eorge( [Ww]\.)? [Bb]ush
```

will match the lines

```
i bet i can spell better than you and george bush combined

BBC reported that President George W. Bush claimed
God told him to invade Iraq

a bird in the hand is worth two george bushes
```

One thing to note...

In the following

```
[Gg]eorge( [Ww]\.)? [Bb]ush
```

we wanted to match a "." as a literal period; to do that, we had to "escape" the metacharacter, preceding it with a backslash In general, we have to do this for any metacharacter we want to include in our match

More metacharacters: * and +

The * and + signs are metacharacters used to indicate repetition; * means "any number, including none, of the item" and + means "at least one of the item"

```
\(.*\)
```

will match the lines

```
anyone wanna chat? (24, m, germany)

hello, 20.m here... ( east area + drives + webcam )

(he means older men)

()
```

More metacharacters: * and +

The * and + signs are metacharacters used to indicate repetition; * means “any number, including none, of the item” and + means “at least one of the item”

```
[0-9]+ (.*)[0-9]+
```

will match the lines

```
working as MP here 720 MP battallion, 42nd birgade

so say 2 or 3 years at colleage and 4 at uni makes us 23 when and if we finish

it went down on several occasions for like, 3 or 4 *days*

Mmmm its time 4 me 2 go 2 bed
```

More metacharacters: { and }

{ and } are referred to as interval quantifiers; the let us specify the minimum and maximum number of matches of an expression

```
[Oo]bama( +[^ ]+){1,5} debate
```

will match the lines

```
why is it every time I hear Newt speak I want to vote for
him? He would crush Obama in a debate

One Gaddafi moment for Obama was worth all the debates
put together!

Romney + Obama should debate using the universal language
of music via that contraption Dick Van Dyke wore in Mary
Poppins

#Obama and #Boehner could settle #debtlimit debate once
and for all w/ an old fashioned staring contest.
```

More metacharacters: { and }

- {m,n} means at least m but not more than n matches
- {m} means exactly m matches
- {m,} means at least m matches
- {,n} means at most n matches

More metacharacters: (and) revisited

- In most implementations of regular expressions, the parentheses not only limit the scope of alternatives divided by a "|", but also can be used to "remember" text matched by the subexpression enclosed
- We refer to the matched text with \1, \2, etc.

More metacharacters: (and) revisited

So the expression

```
 +([a-zA-Z]+) +\1 +
```

will match the lines

```
time for bed, night night twitter!

blah blah blah blah

my tattoo is so so itchy today

i was standing all all alone against the world outside...

hi anybody anybody at home

estudiando css css css css.... que desastritooooo
```

More metacharacters: (and) revisited

The * is “greedy” so it always matches the *longest* possible string that satisfies the regular expression. So

```
^s(.*)s
```

matches

```
sitting at starbucks

setting up mysql and rails

studying stuff for the exams

spaghetti with marshmallows

stop fighting with crackers

sore shoulders, stupid ergonomics
```

More metacharacters: (and) revisited

The greediness of * can be turned off with the ?, as in

```
^s(.*?)s
```

Summary

Regular expressions consist of

- Literals, strings of characters
- Character classes
- General metacharacters, descriptions of complex word/symbol combinations