- There are two types of cells: Markdown cells (like this one) and code cells
- When a cell is selected, pressing `Enter` puts you in editing mode and pressing `Escape` takes you out of editing mode
- Pressing `Shift + Enter` runs the cell and proceeds to the next cell, and `Ctrl + Enter` just runs the cell

Run the following cell:

```
In [1]: 1 + 1
```

```
Out[1]: 2
```

- To change a code cell to a Markdown cell, exit editing mode and press `M`
- To change a Markdown cell to a code cell, enter then exit editing mode, and then press `Y`

- To create a cell above or below the current cell, use `a` and `b` while outside editing mode
- To delete a cell, use `x`

```
In [2]: println("Hi!")
```

```
Hi!
```

Fill in the following cell (which will be placed at the beginning of future exercises):

Last name: Nitski

First name: Osvald

Student number: 1002456987

Submission instructions for when you're done:

Submit your executed notebook as a PDF document according to the instructions in the course syllabus. The most robust way to convert to PDF is as follows:

- Go to File -> Download as -> html and download the notebook as HTML
- Open the HTML document in your browser and print it as a PDF
  - If you don't know how to print to PDF, Google instructions specific to your web browser
- Make sure that all of the required output is visible in the PDF

To check the documentation of a function, use a `?`

Some initial tips:

- Use Kernel -> Interrupt to stop something from running (infinite loop, taking too long, etc.)
- Use Kernel -> Restart or Restart & Clear Output to restart the Julia session clearing all variables, function definitions, etc.
- Consider using Restart & Run All once you've completed your notebook before exporting for submission (make sure any required output is present though!)
- Autosaving isn't continuous so save liberally by pressing `Ctrl + S`

To check the documentation of a function, use `?` as follows:

In [3]:
```
?println
```

```
search: println printstyled print sprint isprint
```

Out[3]:
```
println([io::IO], xs...)
```

Print (using `print` (@ref)) `xs` followed by a newline. If `io` is not supplied, prints to `stdout` (@ref).

Examples

```
jldoctest
julia> println("Hello, world")
Hello, world

julia> io = IOBuffer();

julia> println(io, "Hello, world")

julia> String(take!(io))
"Hello, world\n"
```

Run the following cells:

In [4]:
```
x = 2 + 2
```

Out[4]:
```
4
```

In [5]:
```
y = 9/2
```

Out[5]:
```
4.5
```

In [6]:
```
x^2
```

Out[6]:
```
16
```

In [7]:
```
4%2
```

Out[7]:
```
0
```

In [8]:
```
3%2
```

Out[8]:
```
1
```

In [9]:
```
typeof(x)
```

Out[9]:
```
Int64
```

In [10]:
```
typeof(y)
```

Out[10]:
```
Float64
```

Unicode variable names are supported:

- Use `\mu` followed by a `Tab` to get a μ
- Use `\:smi` follwed by a `Tab`, select the emoji from the list and press `enter`, then press `Tab` followed by `enter`

In [11]:
```
λ = 7
1 + 2λ
```

Out[11]:
```
15
```

In [12]:
```
😢 = 14
```

Out[12]:
```
14
```

In [13]:
```
😄 = 15
```

Out[13]:
```
15
```

In [14]:
```
😄 == 😢 + 1
```

Out[14]:
```
true
```

Generic and typed arrays can be created. Run the following cells:

In [15]:
```
A = [1, 2, 7]
```

Out[15]:
```
3-element Array{Int64,1}:
 1
 2
 7
```

In [16]:
```
B = Float64[1, 2, 7]
```

Out[16]:
```
3-element Array{Float64,1}:
 1.0
 2.0
 7.0
```

In [18]:
```
C = [1, "Two", 7]
```

Out[18]:
```
3-element Array{Any,1}:
 1
  "Two"
 7
```

Indexing is 1-based

In [19]:
```
C[3]
```

Out[19]:
```
7
```

- Copying mutable objects works as in Python
- Comments are inserted using `#`
- The shortcut `Ctrl + /` can be used to comment out a line or multiple highlighted lines

In [20]:
```
A
```

Out[20]:
```
3-element Array{Int64,1}:
 1
 2
 7
```

In [21]:
```
D = A # D refers to A
D[2] = 100; # ; suppresses output
```

Out[21]:
```
100
```

In [22]:
```
D
```

Out[22]:
```
3-element Array{Int64,1}:
   1
 100
   7
```

In [23]:
```
A
```

Out[23]:
```
3-element Array{Int64,1}:
   1
 100
   7
```

In [24]:
```
E = copy(D)
```

Out[24]:
```
3-element Array{Int64,1}:
   1
 100
   7
```

In [25]:
```
E = E.*E
```

Out[25]:
```
3-element Array{Int64,1}:
     1
 10000
    49
```

In [26]:
```
D
```

Out[26]:
```
3-element Array{Int64,1}:
   1
 100
   7
```

In [27]:
```
F = E[:] # Equivalent to copy
```

Out[27]:
```
3-element Array{Int64,1}:
     1
 10000
    49
```

In [28]:
```
F = F .+ 1
```

Out[28]:
```
3-element Array{Int64,1}:
     2
 10001
    50
```

In [29]:
```
E
```

Out[29]:
```
3-element Array{Int64,1}:
     1
 10000
    49
```

- Logical operators are `&&`, `||`, `==`, `!=`, `<`, `<=`, etc.
- Spaces/tabs and colons are not necessary in functions, loops, conditionals, etc.

Traditional function definition:

In [30]:
```
function f(x)
    return x^2
end
```

Out[30]:
```
f (generic function with 1 method)
```

Fancy function definition:

In [31]:
```
g(x) = x^3
```

Out[31]:
```
g (generic function with 1 method)
```

In [32]:
```
f(5) # 5^2
```

Out[32]:
```
25
```

In [33]:
```
g(2) # 2^3
```

Out[33]:
```
8
```

Multiplication concatenates strings

In [34]:
```
"Cheese"*"cake"
```

Out[34]:
```
"Cheesecake"
```

In [35]:
```
M = [1 0; 1 1]
```

Out[35]:
```
2×2 Array{Int64,2}:
 1  0
 1  1
```

In [36]:
```
g(M) # M*M*M
```

Out[36]:
```
2×2 Array{Int64,2}:
 1  0
 3  1
```

In [37]:
```
g("Yo")
```

Out[37]:
```
"YoYoYo"
```

You can also create another variant of the function specific to integers:

In [38]:
```
f(x::Int64) = x*x
```

Out[38]:
```
f (generic function with 2 methods)
```

In [39]:
```
f(5)
```

Out[39]:
```
25
```

`@which` can be used to tell you which variant is called for the argument that you passed

In [40]:
```
@which f(5)
```

Out[40]: f(x::**Int64**) in Main at In[38]:1

- This can be done with different numbers and types of arguments (called multiple dispatch)
- The methods that implement some function can be obtained using `methods()`
- Over 100 methods implement `+`

In [41]:
```
methods(+)
```

Out[41]: 161 methods for generic function **+**:

- +(x::**Bool**, z::**Complex{Bool}**) in Base at complex.jl:278 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/c
- +(x::**Bool**, y::**Bool**) in Base at bool.jl:96 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/b
- +(x::**Bool**) in Base at bool.jl:93 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/b
- +(x::**Bool**, y::**T**) *where T<:AbstractFloat* in Base at bool.jl:104 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/b
- +(x::**Bool**, z::**Complex**) in Base at complex.jl:285 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/c
- +(a::**Float16**, b::**Float16**) in Base at float.jl:392 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/fl
- +(x::**Float32**, y::**Float32**) in Base at float.jl:394 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/fl
- +(x::**Float64**, y::**Float64**) in Base at float.jl:395 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/fl
- +(z::**Complex{Bool}**, x::**Bool**) in Base at complex.jl:279 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/c
- +(z::**Complex{Bool}**, x::**Real**) in Base at complex.jl:293 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/c
- +(::**Missing**, ::**Missing**) in Base at missing.jl:93 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/m
- +(::**Missing**) in Base at missing.jl:79 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/m
- +(::**Missing**, ::**Number**) in Base at missing.jl:94 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/m
- +(level::**Base.CoreLogging.LogLevel**, inc::**Integer**) in Base.CoreLogging at logging.jl:106 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/lo
- +(c::**BigInt**, x::**BigFloat**) in Base.MPFR at mpfr.jl:413 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/m
- +(a::**BigInt**, b::**BigInt**, c::**BigInt**, d::**BigInt**, e::**BigInt**) in Base.GMP at gmp.jl:455 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/g
- +(a::**BigInt**, b::**BigInt**, c::**BigInt**, d::**BigInt**) in Base.GMP at gmp.jl:454 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/g
- +(a::**BigInt**, b::**BigInt**, c::**BigInt**) in Base.GMP at gmp.jl:453 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/g
- +(x::**BigInt**, y::**BigInt**) in Base.GMP at gmp.jl:424 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/g
- +(x::**BigInt**, c::**Union{UInt16, UInt32, UInt8}**) in Base.GMP at gmp.jl:461 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/g
- +(x::**BigInt**, c::**Union{Int16, Int32, Int8}**) in Base.GMP at gmp.jl:467 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/g
- +(a::**BigFloat**, b::**BigFloat**, c::**BigFloat**, d::**BigFloat**, e::**BigFloat**) in Base.MPFR at mpfr.jl:563 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/m
- +(a::**BigFloat**, b::**BigFloat**, c::**BigFloat**, d::**BigFloat**) in Base.MPFR at mpfr.jl:556 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/m

- +(a::**BigFloat**, b::**BigFloat**, c::**BigFloat**) in Base.MPFR at mpfr.jl:550 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/m
- +(x::**BigFloat**, c::**BigInt**) in Base.MPFR at mpfr.jl:409 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/m
- +(x::**BigFloat**, y::**BigFloat**) in Base.MPFR at mpfr.jl:378 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/m
- +(x::**BigFloat**, c::**Union{UInt16, UInt32, UInt8}**) in Base.MPFR at mpfr.jl:385 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/m
- +(x::**BigFloat**, c::**Union{Int16, Int32, Int8}**) in Base.MPFR at mpfr.jl:393 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/m
- +(x::**BigFloat**, c::**Union{Float16, Float32, Float64}**) in Base.MPFR at mpfr.jl:401 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/m
- +(x::**Dates.CompoundPeriod**, y::**Dates.CompoundPeriod**) in Dates at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(x::**Dates.CompoundPeriod**, y::**Dates.Period**) in Dates at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(x::**Dates.CompoundPeriod**, y::**Dates.TimeType**) in Dates at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(x::**Dates.Date**, y::**Dates.Day**) in Dates at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(x::**Dates.Date**, y::**Dates.Week**) in Dates at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(dt::**Dates.Date**, z::**Dates.Month**) in Dates at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(dt::**Dates.Date**, y::**Dates.Year**) in Dates at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(dt::**Dates.Date**, t::**Dates.Time**) in Dates at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(t::**Dates.Time**, dt::**Dates.Date**) in Dates at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(x::**Dates.Time**, y::**Dates.TimePeriod**) in Dates at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(dt::**Dates.DateTime**, z::**Dates.Month**) in Dates at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(dt::**Dates.DateTime**, y::**Dates.Year**) in Dates at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib

- +(x::**Dates.DateTime**, y::**Dates.Period**) in Dates at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(B::**BitArray{2}**, J::**LinearAlgebra.UniformScaling**) in LinearAlgebra at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(a::**Pkg.Resolve.VersionWeights.VersionWeight**, b::**Pkg.Resolve.VersionWeights.VersionWeight**) in Pkg.Resolve.VersionWeights at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(a::**Pkg.Resolve.MaxSum.FieldValues.FieldValue**, b::**Pkg.Resolve.MaxSum.FieldValues.FieldValue**) in Pkg.Resolve.MaxSum.FieldValues at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(y::**AbstractFloat**, x::**Bool**) in Base at bool.jl:106 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/b
- +(x::**T**, y::**T**) *where T<:Union{Int128, Int16, Int32, Int64, Int8, UInt128, UInt16, UInt32, UInt64, UInt8}* in Base at int.jl:53 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/in
- +(c::**Union{UInt16, UInt32, UInt8}**, x::**BigInt**) in Base.GMP at gmp.jl:462 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/g
- +(c::**Union{Int16, Int32, Int8}**, x::**BigInt**) in Base.GMP at gmp.jl:468 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/g
- +(a::**Integer**, b::**Integer**) in Base at int.jl:871 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/in
- +(x::**Integer**, y::**Ptr**) in Base at pointer.jl:161 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/p
- +(z::**Complex**) in Base at complex.jl:265 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/c
- +(z::**Complex**, w::**Complex**) in Base at complex.jl:267 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/c
- +(z::**Complex**, x::**Bool**) in Base at complex.jl:286 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/c
- +(x::**Real**, z::**Complex{Bool}**) in Base at complex.jl:292 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/c
- +(x::**Real**, z::**Complex**) in Base at complex.jl:304 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/c
- +(z::**Complex**, x::**Real**) in Base at complex.jl:305 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/c
- +(x::**Rational**, y::**Rational**) in Base at rational.jl:254 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/r
- +(x::**Integer**, y::**AbstractChar**) in Base at char.jl:224 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/c
- +(c::**Union{UInt16, UInt32, UInt8}**, x::**BigFloat**) in Base.MPFR at mpfr.jl:389 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/m
- +(c::**Union{Int16, Int32, Int8}**, x::**BigFloat**) in Base.MPFR at mpfr.jl:397 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/m
- +(c::**Union{Float16, Float32, Float64}**, x::**BigFloat**) in Base.MPFR at mpfr.jl:405 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/m

- +(x::**AbstractIrrational**, y::**AbstractIrrational**) in Base at irrationals.jl:137 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/ir
- +(x::**Number**) in Base at operators.jl:504 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/c
- +(x::**T**, y::**T**) *where T<:Number* in Base at promotion.jl:389 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/p
- +(x::**Number**, y::**Number**) in Base at promotion.jl:313 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/p
- +(r1::**OrdinalRange**, r2::**OrdinalRange**) in Base at range.jl:1004 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/r
- +(r1::**LinRange{T}**, r2::**LinRange{T}**) *where T* in Base at range.jl:1011 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/r
- +(r1::**StepRangeLen{T,R,S} where S**, r2::**StepRangeLen{T,R,S} where S**) *where {R<:Base.TwicePrecision, T}* in Base at twiceprecision.jl:557 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/tw
- +(r1::**StepRangeLen{T,S,S1} where S1**, r2::**StepRangeLen{T,S,S1} where S1**) *where {T, S}* in Base at range.jl:1027 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/r
- +(r1::**Union{LinRange, OrdinalRange, StepRangeLen}**, r2::**Union{LinRange, OrdinalRange, StepRangeLen}**) in Base at range.jl:1019 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/r
- +(x::**Ptr**, y::**Integer**) in Base at pointer.jl:159 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/p
- +(x::**Base.TwicePrecision**, y::**Number**) in Base at twiceprecision.jl:265 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/tw
- +(x::**Number**, y::**Base.TwicePrecision**) in Base at twiceprecision.jl:268 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/tw
- +(x::**Base.TwicePrecision{T}**, y::**Base.TwicePrecision{T}**) *where T* in Base at twiceprecision.jl:271 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/tw
- +(x::**Base.TwicePrecision**, y::**Base.TwicePrecision**) in Base at twiceprecision.jl:275 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/tw
- +(A::**Array**, Bs::**Array...**) in Base at arraymath.jl:44 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/a
- +(A::**BitArray**, B::**BitArray**) in Base at bitarray.jl:1084 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/b
- +(r::**AbstractRange{#s617} where #s617<:Dates.TimeType**, x::**Dates.Period**) in Dates at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(A::**Array**, B::**SparseArrays.SparseMatrixCSC**) in SparseArrays at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(x::**Union{DenseArray{#s617,N}, Base.ReinterpretArray{#s617,N,S,A} where S where A<:Union{SubArray{T,N,A,I,true} where I<:Union{Tuple{Vararg{Real,N} where N}, Tuple{AbstractUnitRange,Vararg{Any,N} where N}} where A<:DenseArray where N where T, DenseArray}, Base.ReshapedArray{#s617,N,A,MI} where MI<:Tuple{Vararg{Base.MultiplicativeInverses.SignedMultiplicativeInverse{Int64},N} where N} where A<:Union{Base.ReinterpretArray{T,N,S,A} where S where A<:Union{SubArray{T,N,A,I,true} where I<:Union{Tuple{Vararg{Real,N} where N},**

Tuple{AbstractUnitRange,Vararg{Any,N} where N}} where A<:DenseArray where N where T, DenseArray} where N where T, SubArray{T,N,A,I,true} where I<:Union{Tuple{Vararg{Real,N} where N}, Tuple{AbstractUnitRange,Vararg{Any,N} where N}} where A<:DenseArray where N where T, DenseArray}, SubArray{#s617,N,A,I,L} where L where I<:Tuple{Vararg{Union{Int64, AbstractRange{Int64}, Base.AbstractCartesianIndex},N} where N} where A<:Union{Base.ReinterpretArray{T,N,S,A} where S where A<:Union{SubArray{T,N,A,I,true} where I<:Union{Tuple{Vararg{Real,N} where N}, Tuple{AbstractUnitRange,Vararg{Any,N} where N}} where A<:DenseArray where N where T, DenseArray} where N where T, Base.ReshapedArray{T,N,A,MI} where MI<:Tuple{Vararg{Base.MultiplicativeInverses.SignedMultiplicativeInverse{Int64},N} where N} where A<:Union{Base.ReinterpretArray{T,N,S,A} where S where A<:Union{SubArray{T,N,A,I,true} where I<:Union{Tuple{Vararg{Real,N} where N}, Tuple{AbstractUnitRange,Vararg{Any,N} where N}} where A<:DenseArray where N where T, DenseArray} where N where T, SubArray{T,N,A,I,true} where I<:Union{Tuple{Vararg{Real,N} where N}, Tuple{AbstractUnitRange,Vararg{Any,N} where N}} where A<:DenseArray where N where T, DenseArray} where N where T, DenseArray}} where N where #s617<:Union{Dates.CompoundPeriod, Dates.Period}) in Dates at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib

- +(x::**Union{DenseArray{#s617,N}, Base.ReinterpretArray{#s617,N,S,A} where S where A<:Union{SubArray{T,N,A,I,true} where I<:Union{Tuple{Vararg{Real,N} where N}, Tuple{AbstractUnitRange,Vararg{Any,N} where N}} where A<:DenseArray where N where T, DenseArray}, Base.ReshapedArray{#s617,N,A,MI} where MI<:Tuple{Vararg{Base.MultiplicativeInverses.SignedMultiplicativeInverse{Int64},N} where N} where A<:Union{Base.ReinterpretArray{T,N,S,A} where S where A<:Union{SubArray{T,N,A,I,true} where I<:Union{Tuple{Vararg{Real,N} where N}, Tuple{AbstractUnitRange,Vararg{Any,N} where N}} where A<:DenseArray where N where T, DenseArray} where N where T, SubArray{T,N,A,I,true} where I<:Union{Tuple{Vararg{Real,N} where N}, Tuple{AbstractUnitRange,Vararg{Any,N} where N}} where A<:DenseArray where N where T, DenseArray}, SubArray{#s617,N,A,I,L} where L where I<:Tuple{Vararg{Union{Int64, AbstractRange{Int64}, Base.AbstractCartesianIndex},N} where N} where A<:Union{Base.ReinterpretArray{T,N,S,A} where S where A<:Union{SubArray{T,N,A,I,true} where I<:Union{Tuple{Vararg{Real,N} where N}, Tuple{AbstractUnitRange,Vararg{Any,N} where N}} where A<:DenseArray where N where T, DenseArray} where N where T, Base.ReshapedArray{T,N,A,MI} where MI<:Tuple{Vararg{Base.MultiplicativeInverses.SignedMultiplicativeInverse{Int64},N} where N} where A<:Union{Base.ReinterpretArray{T,N,S,A} where S where A<:Union{SubArray{T,N,A,I,true} where I<:Union{Tuple{Vararg{Real,N} where N}, Tuple{AbstractUnitRange,Vararg{Any,N} where N}} where A<:DenseArray where N where T, DenseArray} where N where T, SubArray{T,N,A,I,true} where I<:Union{Tuple{Vararg{Real,N} where N}, Tuple{AbstractUnitRange,Vararg{Any,N} where N}} where A<:DenseArray where N where T, DenseArray} where N where T, DenseArray}} where N where #s617<:Union{Dates.CompoundPeriod, Dates.Period}**, y::**Dates.TimeType**) in Dates at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib

- +(X::**Union{DenseArray{#s617,N}, Base.ReinterpretArray{#s617,N,S,A} where S where A<:Union{SubArray{T,N,A,I,true} where I<:Union{Tuple{Vararg{Real,N} where N}, Tuple{AbstractUnitRange,Vararg{Any,N} where N}} where A<:DenseArray where N where T, DenseArray}, Base.ReshapedArray{#s617,N,A,MI} where MI<:Tuple{Vararg{Base.MultiplicativeInverses.SignedMultiplicativeInverse{Int64},N} where N} where A<:Union{Base.ReinterpretArray{T,N,S,A} where S where A<:Union{SubArray{T,N,A,I,true} where I<:Union{Tuple{Vararg{Real,N} where N}, Tuple{AbstractUnitRange,Vararg{Any,N} where N}} where A<:DenseArray where N where T, DenseArray} where N where T, SubArray{T,N,A,I,true} where I<:Union{Tuple{Vararg{Real,N} where N}, Tuple{AbstractUnitRange,Vararg{Any,N} where N}} where A<:DenseArray where N where T, DenseArray}, SubArray{#s617,N,A,I,L} where L where I<:Tuple{Vararg{Union{Int64, AbstractRange{Int64}, Base.AbstractCartesianIndex},N} where N} where A<:Union{Base.ReinterpretArray{T,N,S,A} where S where A<:Union{SubArray{T,N,A,I,true} where I<:Union{Tuple{Vararg{Real,N} where N}, Tuple{AbstractUnitRange,Vararg{Any,N} where N}} where A<:DenseArray where N where T, DenseArray} where N where T, Base.ReshapedArray{T,N,A,MI} where MI<:Tuple{Vararg{Base.MultiplicativeInverses.SignedMultiplicativeInverse{Int64},N} where N} where A<:Union{Base.ReinterpretArray{T,N,S,A} where S where A<:Union{SubArray{T,N,A,I,true} where I<:Union{Tuple{Vararg{Real,N} where N}, Tuple{AbstractUnitRange,Vararg{Any,N} where N}} where A<:DenseArray where N where T, DenseArray} where N where T, SubArray{T,N,A,I,true} where I<:Union{Tuple{Vararg{Real,N} where N}, Tuple{AbstractUnitRange,Vararg{Any,N} where N}} where A<:DenseArray where N where T, DenseArray} where N where T, DenseArray}} where N where #s617<:Union{Dates.CompoundPeriod, Dates.Period}**, Y::**Union{DenseArray{#s616,N}, Base.ReinterpretArray{#s616,N,S,A} where S where A<:Union{SubArray{T,N,A,I,true} where I<:Union{Tuple{Vararg{Real,N} where N}, Tuple{AbstractUnitRange,Vararg{Any,N} where N}} where A<:DenseArray where N where T, DenseArray}, Base.ReshapedArray{#s616,N,A,MI} where MI<:Tuple{Vararg{Base.MultiplicativeInverses.SignedMultiplicativeInverse{Int64},N} where N} where A<:Union{Base.ReinterpretArray{T,N,S,A} where S where A<:Union{SubArray{T,N,A,I,true} where I<:Union{Tuple{Vararg{Real,N} where N}, Tuple{AbstractUnitRange,Vararg{Any,N} where N}} where A<:DenseArray where N where T, DenseArray} where N where T, SubArray{T,N,A,I,true} where I<:Union{Tuple{Vararg{Real,N} where N}, Tuple{AbstractUnitRange,Vararg{Any,N} where N}} where A<:DenseArray where N where T, DenseArray}, SubArray{#s616,N,A,I,L} where L where I<:Tuple{Vararg{Union{Int64, AbstractRange{Int64}, Base.AbstractCartesianIndex},N} where N} where A<:Union{Base.ReinterpretArray{T,N,S,A} where S where A<:Union{SubArray{T,N,A,I,true} where I<:Union{Tuple{Vararg{Real,N} where N}, Tuple{AbstractUnitRange,Vararg{Any,N} where N}} where A<:DenseArray where N where T, DenseArray} where N where T, Base.ReshapedArray{T,N,A,MI} where MI<:Tuple{Vararg{Base.MultiplicativeInverses.SignedMultiplicativeInverse{Int64},N} where N} where A<:Union{Base.ReinterpretArray{T,N,S,A} where S where A<:Union{SubArray{T,N,A,I,true} where I<:Union{Tuple{Vararg{Real,N} where N}, Tuple{AbstractUnitRange,Vararg{Any,N} where N}} where A<:DenseArray where N where T, DenseArray} where N where T, SubArray{T,N,A,I,true} where I<:Union{Tuple{Vararg{Real,N} where N}, Tuple{AbstractUnitRange,Vararg{Any,N} where N}} where A<:DenseArray where N where T, DenseArray} where N where T,**

DenseArray}} where N where #s616<:Union{Dates.CompoundPeriod, Dates.Period}) in Dates at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib

- +(A::**LinearAlgebra.SymTridiagonal**, B::**LinearAlgebra.SymTridiagonal**) in LinearAlgebra at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(A::**LinearAlgebra.Tridiagonal**, B::**LinearAlgebra.Tridiagonal**) in LinearAlgebra at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(A::**LinearAlgebra.UpperTriangular**, B::**LinearAlgebra.UpperTriangular**) in LinearAlgebra at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(A::**LinearAlgebra.LowerTriangular**, B::**LinearAlgebra.LowerTriangular**) in LinearAlgebra at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(A::**LinearAlgebra.UpperTriangular**, B::**LinearAlgebra.UnitUpperTriangular**) in LinearAlgebra at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(A::**LinearAlgebra.LowerTriangular**, B::**LinearAlgebra.UnitLowerTriangular**) in LinearAlgebra at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(A::**LinearAlgebra.UnitUpperTriangular**, B::**LinearAlgebra.UpperTriangular**) in LinearAlgebra at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(A::**LinearAlgebra.UnitLowerTriangular**, B::**LinearAlgebra.LowerTriangular**) in LinearAlgebra at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(A::**LinearAlgebra.UnitUpperTriangular**, B::**LinearAlgebra.UnitUpperTriangular**) in LinearAlgebra at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(A::**LinearAlgebra.UnitLowerTriangular**, B::**LinearAlgebra.UnitLowerTriangular**) in LinearAlgebra at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(A::**LinearAlgebra.AbstractTriangular**, B::**LinearAlgebra.AbstractTriangular**) in LinearAlgebra at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(A::**LinearAlgebra.Symmetric**, B::**LinearAlgebra.Symmetric**) in LinearAlgebra at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2

(file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib

- +(A::**LinearAlgebra.Hermitian**, B::**LinearAlgebra.Hermitian**) in LinearAlgebra at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(A::**LinearAlgebra.Hermitian**, B::**LinearAlgebra.Symmetric{#s617,S} where S<:(AbstractArray{#s6171,2} where #s6171<:#s617) where #s617<:Real**) in LinearAlgebra at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(A::**LinearAlgebra.Symmetric{#s617,S} where S<:(AbstractArray{#s6171,2} where #s6171<:#s617) where #s617<:Real**, B::**LinearAlgebra.Hermitian**) in LinearAlgebra at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(Da::**LinearAlgebra.Diagonal**, Db::**LinearAlgebra.Diagonal**) in LinearAlgebra at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(A::**LinearAlgebra.Bidiagonal**, B::**LinearAlgebra.Bidiagonal**) in LinearAlgebra at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(UL::**LinearAlgebra.UnitUpperTriangular**, J::**LinearAlgebra.UniformScaling**) in LinearAlgebra at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(UL::**LinearAlgebra.UnitLowerTriangular**, J::**LinearAlgebra.UniformScaling**) in LinearAlgebra at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(A::**LinearAlgebra.Hermitian**, J::**LinearAlgebra.UniformScaling{#s617} where #s617<:Complex**) in LinearAlgebra at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(A::**LinearAlgebra.Tridiagonal{#s617,V} where V<:AbstractArray{#s617,1} where #s617<:Number**, B::**LinearAlgebra.UniformScaling**) in LinearAlgebra at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(A::**LinearAlgebra.SymTridiagonal{#s617,V} where V<:AbstractArray{#s617,1} where #s617<:Number**, B::**LinearAlgebra.UniformScaling**) in LinearAlgebra at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(A::**LinearAlgebra.Bidiagonal{#s617,V} where V<:AbstractArray{#s617,1} where #s617<:Number**, B::**LinearAlgebra.UniformScaling**) in LinearAlgebra at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(A::**LinearAlgebra.Diagonal{#s617,V} where V<:AbstractArray{#s617,1} where #s617<:Number**, B::**LinearAlgebra.UniformScaling**) in LinearAlgebra at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(A::**SparseArrays.SparseMatrixCSC**, J::**LinearAlgebra.UniformScaling**) in SparseArrays at

C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2
(file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib

- +(A::**AbstractArray{T,2} where T**, J::**LinearAlgebra.UniformScaling**) in LinearAlgebra at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(A::**LinearAlgebra.UpperTriangular**, B::**LinearAlgebra.Bidiagonal**) in LinearAlgebra at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(A::**LinearAlgebra.Bidiagonal**, B::**LinearAlgebra.UpperTriangular**) in LinearAlgebra at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(A::**LinearAlgebra.UnitUpperTriangular**, B::**LinearAlgebra.Bidiagonal**) in LinearAlgebra at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(A::**LinearAlgebra.Bidiagonal**, B::**LinearAlgebra.UnitUpperTriangular**) in LinearAlgebra at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(A::**LinearAlgebra.LowerTriangular**, B::**LinearAlgebra.Bidiagonal**) in LinearAlgebra at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(A::**LinearAlgebra.Bidiagonal**, B::**LinearAlgebra.LowerTriangular**) in LinearAlgebra at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(A::**LinearAlgebra.UnitLowerTriangular**, B::**LinearAlgebra.Bidiagonal**) in LinearAlgebra at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(A::**LinearAlgebra.Bidiagonal**, B::**LinearAlgebra.UnitLowerTriangular**) in LinearAlgebra at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(A::**LinearAlgebra.Bidiagonal**, B::**LinearAlgebra.Diagonal**) in LinearAlgebra at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(A::**LinearAlgebra.Diagonal**, B::**LinearAlgebra.Bidiagonal**) in LinearAlgebra at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(A::**LinearAlgebra.Diagonal**, B::**LinearAlgebra.SymTridiagonal**) in LinearAlgebra at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(A::**LinearAlgebra.SymTridiagonal**, B::**LinearAlgebra.Diagonal**) in LinearAlgebra at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(A::**LinearAlgebra.Tridiagonal**, B::**LinearAlgebra.SymTridiagonal**) in LinearAlgebra at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib

- +(A::**LinearAlgebra.SymTridiagonal**, B::**LinearAlgebra.Tridiagonal**) in LinearAlgebra at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(A::**LinearAlgebra.Diagonal**, B::**LinearAlgebra.Tridiagonal**) in LinearAlgebra at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(A::**LinearAlgebra.Tridiagonal**, B::**LinearAlgebra.Diagonal**) in LinearAlgebra at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(A::**LinearAlgebra.Bidiagonal**, B::**LinearAlgebra.Tridiagonal**) in LinearAlgebra at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(A::**LinearAlgebra.Tridiagonal**, B::**LinearAlgebra.Bidiagonal**) in LinearAlgebra at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(A::**LinearAlgebra.Bidiagonal**, B::**LinearAlgebra.SymTridiagonal**) in LinearAlgebra at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(A::**LinearAlgebra.SymTridiagonal**, B::**LinearAlgebra.Bidiagonal**) in LinearAlgebra at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(A::**SparseArrays.SparseMatrixCSC**, B::**SparseArrays.SparseMatrixCSC**) in SparseArrays at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(A::**SparseArrays.SparseMatrixCSC**, B::**Array**) in SparseArrays at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(x::**SparseArrays.AbstractSparseArray{Tv,Ti,1} where Ti where Tv**, y::**SparseArrays.AbstractSparseArray{Tv,Ti,1} where Ti where Tv**) in SparseArrays at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(x::**AbstractArray{#s75,N} where N where #s75<:Number**) in Base at abstractarraymath.jl:97 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/a
- +(A::**AbstractArray**, B::**AbstractArray**) in Base at arraymath.jl:38 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/a
- +(x::**T**, y::**Integer**) *where T<:AbstractChar* in Base at char.jl:223 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/c
- +(index1::**CartesianIndex{N}**, index2::**CartesianIndex{N}**) *where N* in Base.IteratorsMD at multidimensional.jl:110 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/m
- +(::**Number**, ::**Missing**) in Base at missing.jl:95 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/m
- +(x::**P**, y::**P**) *where P<:Dates.Period* in Dates at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(x::**Dates.Period**, y::**Dates.Period**) in Dates at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2

(file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib

- +(y::**Dates.Period**, x::**Dates.CompoundPeriod**) in Dates at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(x::**Union{Dates.CompoundPeriod, Dates.Period}**) in Dates at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(x::**Dates.TimeType**) in Dates at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(a::**Dates.TimeType**, b::**Dates.Period**, c::**Dates.Period**) in Dates at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(a::**Dates.TimeType**, b::**Dates.Period**, c::**Dates.Period**, d::**Dates.Period...**) in Dates at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(x::**Dates.TimeType**, y::**Dates.CompoundPeriod**) in Dates at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(x::**Dates.Instant**) in Dates at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(y::**Dates.Period**, x::**Dates.TimeType**) in Dates at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(x::**Dates.Period**, r::**AbstractRange{#s617} where #s617<:Dates.TimeType**) in Dates at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(x::**AbstractArray{#s617,N} where N where #s617<:Dates.TimeType**, y::**Union{Dates.CompoundPeriod, Dates.Period}**) in Dates at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(y::**Union{Dates.CompoundPeriod, Dates.Period}**, x::**AbstractArray{#s617,N} where N where #s617<:Dates.TimeType**) in Dates at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(y::**Dates.TimeType**, x::**Union{DenseArray{#s617,N}, Base.ReinterpretArray{#s617,N,S,A} where S where A<:Union{SubArray{T,N,A,I,true} where I<:Union{Tuple{Vararg{Real,N} where N}, Tuple{AbstractUnitRange,Vararg{Any,N} where N}} where A<:DenseArray where N where T, DenseArray}, Base.ReshapedArray{#s617,N,A,MI} where MI<:Tuple{Vararg{Base.MultiplicativeInverses.SignedMultiplicativeInverse{Int64},N} where N} where A<:Union{Base.ReinterpretArray{T,N,S,A} where S where A<:Union{SubArray{T,N,A,I,true} where I<:Union{Tuple{Vararg{Real,N} where N}, Tuple{AbstractUnitRange,Vararg{Any,N} where N}} where A<:DenseArray where N where T, DenseArray} where N where T, SubArray{T,N,A,I,true} where I<:Union{Tuple{Vararg{Real,N} where N}, Tuple{AbstractUnitRange,Vararg{Any,N} where N}} where A<:DenseArray where N where T, DenseArray}, SubArray{#s617,N,A,I,L} where L where I<:Tuple{Vararg{Union{Int64,**

AbstractRange{Int64}, Base.AbstractCartesianIndex},N} where N} where A<:Union{Base.ReinterpretArray{T,N,S,A} where S where A<:Union{SubArray{T,N,A,I,true} where I<:Union{Tuple{Vararg{Real,N} where N}, Tuple{AbstractUnitRange,Vararg{Any,N} where N}} where A<:DenseArray where N where T, DenseArray} where N where T, Base.ReshapedArray{T,N,A,MI} where MI<:Tuple{Vararg{Base.MultiplicativeInverses.SignedMultiplicativeInverse{Int64},N} where N} where A<:Union{Base.ReinterpretArray{T,N,S,A} where S where A<:Union{SubArray{T,N,A,I,true} where I<:Union{Tuple{Vararg{Real,N} where N}, Tuple{AbstractUnitRange,Vararg{Any,N} where N}} where A<:DenseArray where N where T, DenseArray} where N where T, SubArray{T,N,A,I,true} where I<:Union{Tuple{Vararg{Real,N} where N}, Tuple{AbstractUnitRange,Vararg{Any,N} where N}} where A<:DenseArray where N where T, DenseArray} where N where T, DenseArray}} where N where #s617<:Union{Dates.CompoundPeriod, Dates.Period}) in Dates at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib

- +(J::**LinearAlgebra.UniformScaling**, x::**Number**) in LinearAlgebra at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(x::**Number**, J::**LinearAlgebra.UniformScaling**) in LinearAlgebra at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(J1::**LinearAlgebra.UniformScaling**, J2::**LinearAlgebra.UniformScaling**) in LinearAlgebra at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(J::**LinearAlgebra.UniformScaling**, B::**BitArray{2}**) in LinearAlgebra at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(A::**LinearAlgebra.UniformScaling**, B::**LinearAlgebra.Tridiagonal{#s617,V} where V<:AbstractArray{#s617,1} where #s617<:Number**) in LinearAlgebra at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(A::**LinearAlgebra.UniformScaling**, B::**LinearAlgebra.SymTridiagonal{#s617,V} where V<:AbstractArray{#s617,1} where #s617<:Number**) in LinearAlgebra at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(A::**LinearAlgebra.UniformScaling**, B::**LinearAlgebra.Bidiagonal{#s617,V} where V<:AbstractArray{#s617,1} where #s617<:Number**) in LinearAlgebra at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(A::**LinearAlgebra.UniformScaling**, B::**LinearAlgebra.Diagonal{#s617,V} where V<:AbstractArray{#s617,1} where #s617<:Number**) in LinearAlgebra at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib
- +(J::**LinearAlgebra.UniformScaling**, A::**AbstractArray{T,2} where T**) in LinearAlgebra at C:\cygwin\home\Administrator\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.2 (file://C:/cygwin/home/Administrator/buildbot/worker/package_win64/build/usr/share/julia/stdlib

- +(a, b, c, xs...) in Base at operators.jl:529 (https://github.com/JuliaLang/julia/tree/c6da87ff4bc7a855e217856757ad3413cf6d1f79/base/c

Array comprehension:

In [42]:
```
A = [i + j for i in 1:3, j in 1:3]
```

Out[42]:
```
3×3 Array{Int64,2}:
 2  3  4
 3  4  5
 4  5  6
```

Counting the number of positive elements in a vector:

In [43]:
```
u = rand(10,1) .- 0.5 # Vector of 10 random numbers between -0.5 and 0.5
```

Out[43]:
```
10×1 Array{Float64,2}:
 -0.3775267186433242
 -0.1397990412058392
  0.44032802226371515
 -0.2387886276898965
 -0.4864233741480277
  0.19701265237143772
  0.13965870149601267
  0.32547604272615205
  0.2025642991970733
 -0.04757721968463691
```

In [44]:
```
u .> 0 # Elementwise logical
```

Out[44]:
```
10×1 BitArray{2}:
 0
 0
 1
 0
 0
 1
 1
 1
 1
 0
```

In [45]:
```
sum(u .> 0) # Number of positive elements
```

Out[45]:
```
5
```

In [46]:
```
v = [i for i in 1:5]
```

Out[46]:
```
5-element Array{Int64,1}:
 1
 2
 3
 4
 5
```

Functions are applied elementwise with a dot as well:

In [47]:
```
f.(v)
```

Out[47]:
```
5-element Array{Int64,1}:
  1
  4
  9
 16
 25
```

Conditionals:

In [48]:
```
function lousyMax(a, b, c)
    if a >= b
        if a >= c
            return a
        else
            return c
        end
    elseif b >= c
        return b
    else
        return c
    end
end
```

Out[48]:
```
lousyMax (generic function with 1 method)
```

In [49]:
```
println(lousyMax(3,7,2))
println(lousyMax(1,6,9))
```

```
7
9
```

Using `$` in a string evaluates the expression that follows and converts it into a string

In [50]:
```
N = 1
while N <= 5
    println("$N squared is $(f(N))")
    N = N + 1
end
```

```
1 squared is 1
2 squared is 4
3 squared is 9
4 squared is 16
5 squared is 25
```

Exercise (Project Euler Problem 2): consider the Fibonacci numbers $F(n)$ which are $1, 2, 3, 5, 8, 13, 21, \ldots$, specifically,

$$F(1) = 1,\ F(2) = 2,\ F(n) = F(n-1) + F(n-2) \quad \forall n \geq 2$$

Compute the sum of the even Fibonacci numbers strictly smaller than four million and store the result in a variable named `fibsum`. Write the code in between this cell and the cell with the `@assert` which verfies your result. Recall that you can use `x`, `a`, and `b` to create and delete cells (after pressing `Escape` to exit edit mode).

In [54]:
```
function F_even_sum(n)
    one_back = 1
    current = 2
    f_sum = 0
    while current < n
        if (current % 2) == 0
            f_sum += current
        end
        two_back = one_back
        one_back = current
        current = one_back + two_back
    end
    return f_sum
end
```

Out[54]: F_even_sum (generic function with 1 method)

In [55]:
```
F_even_sum(20)
```

Out[55]: 10

In [56]:
```
fibsum = F_even_sum(4000000)
```

Out[56]: 4613732

In [57]:
```
@assert fibsum == 4613732 # Throws an error if your answer is wrong
```

Matrices (2-dimensional arrays):

In [58]:
```
A = [1 2 3; 4 5 6; 7 8 9]
```

Out[58]:
```
3×3 Array{Int64,2}:
 1  2  3
 4  5  6
 7  8  9
```

In [59]:
```
size(A)
```

Out[59]:
```
(3, 3)
```

Every one-dimensional array will by default be a "column" vector

In [60]:
```
A[2,:] # Second ROW of A
```

Out[60]:
```
3-element Array{Int64,1}:
 4
 5
 6
```

In [61]:
```
A[:,3] # Third COLUMN of A
```

Out[61]:
```
3-element Array{Int64,1}:
 3
 6
 9
```

In [62]:
```
A[2:3,2:3]
```

Out[62]:
```
2×2 Array{Int64,2}:
 5  6
 8  9
```

In [63]:
```
B = [A 2A; 3A 4A] # You can construct matrices from other matrices
```

Out[63]:
```
6×6 Array{Int64,2}:
  1   2   3   2   4   6
  4   5   6   8  10  12
  7   8   9  14  16  18
  3   6   9   4   8  12
 12  15  18  16  20  24
 21  24  27  28  32  36
```

In [64]:
```
a = Float64[] # Create empty array of floats
```

Out[64]:
```
0-element Array{Float64,1}
```

In [65]:
```
push!(a, 21)
push!(a, 23)
push!(a, 29)
```

Out[65]:
```
3-element Array{Float64,1}:
 21.0
 23.0
 29.0
```

In [66]:
```
a
```

Out[66]:
```
3-element Array{Float64,1}:
 21.0
 23.0
 29.0
```

In [67]:
```
pop!(a)
```

Out[67]:
```
29.0
```

In [68]:
```
a
```

Out[68]:
```
2-element Array{Float64,1}:
 21.0
 23.0
```

In [69]:
```
popfirst!(a)
```

Out[69]:
```
21.0
```

In [70]:
```
a
```

Out[70]:
```
1-element Array{Float64,1}:
 23.0
```

Some useful functions:

In [71]:
```
using LinearAlgebra
```

In [72]:
```
I(3) # Identity matrix
```

Out[72]:
```
3×3 Diagonal{Bool,Array{Bool,1}}:
 1  ·  ·
 ·  1  ·
 ·  ·  1
```

In [73]:
```
A + I(3)
```

Out[73]:
```
3×3 Array{Int64,2}:
 2  2   3
 4  6   6
 7  8  10
```

In [74]:
```
ones(3,3)
```

Out[74]:
```
3×3 Array{Float64,2}:
 1.0  1.0  1.0
 1.0  1.0  1.0
 1.0  1.0  1.0
```

In [75]:
```
A + ones(Int64, 3, 3)
```

Out[75]:
```
3×3 Array{Int64,2}:
 2  3   4
 5  6   7
 8  9  10
```

In [76]:
```
A = rand(3,3)
```

Out[76]:
```
3×3 Array{Float64,2}:
 0.796795  0.699497  0.504204
 0.641303  0.915737  0.67238
 0.347107  0.63302   0.525021
```

In [77]:
```
diag(A)
```

Out[77]:
```
3-element Array{Float64,1}:
 0.796794669956221
 0.9157366833391978
 0.525020604542175
```

In [78]:
```
diagm(diag(A))
```

Out[78]:
```
3×3 Array{Float64,2}:
 0.796795  0.0       0.0
 0.0       0.915737  0.0
 0.0       0.0       0.525021
```

In [79]:
```
inv(A)*A
```

Out[79]:
```
3×3 Array{Float64,2}:
  1.0          -5.55112e-16  -1.66533e-16
 -8.88178e-16   1.0           0.0
  8.88178e-16   0.0           1.0
```

`transpose()` gives the transpose and `'` gives the conjugate tranpose (adjoint, or complex conjugate of transpose)

In [80]:
```
u = rand(3)
v = rand(3)
```

Out[80]:
```
3-element Array{Float64,1}:
 0.6970364028329228
 0.6674618543450423
 0.6559808544942687
```

In [81]:
```
transpose(u)*v # inner product
```

Out[81]:
```
0.7148760963869712
```

In [82]:
```
u'*v
```

Out[82]:
```
0.7148760963869712
```

In [83]:
```
u'v
```

Out[83]:
```
0.7148760963869712
```

In [84]:
```
dot(u,v)
```

Out[84]:
```
0.7148760963869712
```

A two-dimensional rotation matrix is given by:

$$R(\theta) = \begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix}$$

We can define a function that produces it as follows:

In [85]:
```
R(θ) = [cos(θ) -sin(θ); sin(θ) cos(θ)]
```

Out[85]:
```
R (generic function with 1 method)
```

In [86]:
```
R(pi/4)
```

Out[86]:
```
2×2 Array{Float64,2}:
 0.707107  -0.707107
 0.707107   0.707107
```

A two-dimensional shear matrix is given by:

$$S(\lambda) = \begin{bmatrix} 1 & \lambda \\ 0 & 1 \end{bmatrix}$$

Exercise: Define a function which computes the shear matrix as a function of λ:

In [87]:
```
S(λ) = [1 λ; 0 1]
```

Out[87]:
```
S (generic function with 1 method)
```

In [88]:
```
S(4)
```

Out[88]:
```
2×2 Array{Int64,2}:
 1  4
 0  1
```

In [89]:
```
@assert S(4) ==  [1 4; 0 1]
```

Install the standard plotting package by running the following cell:

In [90]:
```
using Pkg;
Pkg.add("Plots")
```

```
  Updating registry at `C:\Users\osval\.julia\registries\General`
  Updating git-repo `https://github.com/JuliaRegistries/General.git`
[1mFetching: [========================================>]  99.9 %0.0 % %3.7 %>
]  8.3 %Fetching: [====>                                    ]  9.8 %>
]  12.3 %                                 ]  13.9 %7 %]  17.9 %
]  20.3 %                               ]  22.4 %23.5 %>
]  26.0 %5 %Fetching: [=============>                          ]  31.1 %33.3
%===============>                        ]  35.3 %]  37.3 %
]  40.4 %]  42.5 %>                      ]  44.6 %>                    ]  4
7.0 %.9 %53.6 % [=======================>                ]  55.7 % [========
================>                ]  57.6 %.7 %8 %>             ]  64.1 % [==
=========================>            ]  66.0 % % %=========================
====>           ]  71.3 % [==============================>          ]  73.3
 %.2 %76.9 %.1 %==================================>       ]  81.2 %3 %.5 % [==
==================================>    ]  87.6 %]  89.5 % %Fetching: [=======
================================> ]  95.1 %]  98.4 % Resolving package versio
ns...
 Installed SortingAlgorithms ── v0.3.1
 Installed Requires ─────────── v0.5.2
 Installed Missings ─────────── v0.4.2
 Installed FFMPEG ───────────── v0.2.3
 Installed Plots ────────────── v0.26.3
 Installed StaticArrays ─────── v0.11.0
 Installed Colors ───────────── v0.9.6
 Installed NaNMath ──────────── v0.3.2
 Installed RecipesBase ──────── v0.7.0
 Installed Reexport ─────────── v0.2.0
 Installed PlotThemes ───────── v0.3.0
 Installed ColorTypes ───────── v0.8.0
 Installed StatsBase ────────── v0.32.0
 Installed OrderedCollections ─ v1.1.0
 Installed Showoff ──────────── v0.3.1
 Installed Measures ─────────── v0.3.0
 Installed FixedPointNumbers ── v0.6.1
 Installed IterTools ────────── v1.2.0
 Installed PlotUtils ────────── v0.5.8
 Installed Contour ──────────── v0.5.1
 Installed DataStructures ───── v0.17.0
 Installed GR ───────────────── v0.41.0
 Installed DataAPI ──────────── v1.0.1
 Installed GeometryTypes ────── v0.7.6
  Updating `C:\Users\osval\.julia\environments\v1.2\Project.toml`
  [91a5bcdd] + Plots v0.26.3
  Updating `C:\Users\osval\.julia\environments\v1.2\Manifest.toml`
  [3da002f7] + ColorTypes v0.8.0
  [5ae59095] + Colors v0.9.6
  [d38c429a] + Contour v0.5.1
  [9a962f9c] + DataAPI v1.0.1
  [864edb3b] + DataStructures v0.17.0
  [c87230d0] + FFMPEG v0.2.3
  [53c48c17] + FixedPointNumbers v0.6.1
  [28b8d3ca] + GR v0.41.0
  [4d00f742] + GeometryTypes v0.7.6
  [c8e1da08] + IterTools v1.2.0
  [442fdcdd] + Measures v0.3.0
  [e1d29d7a] + Missings v0.4.2
  [77ba4419] + NaNMath v0.3.2
```

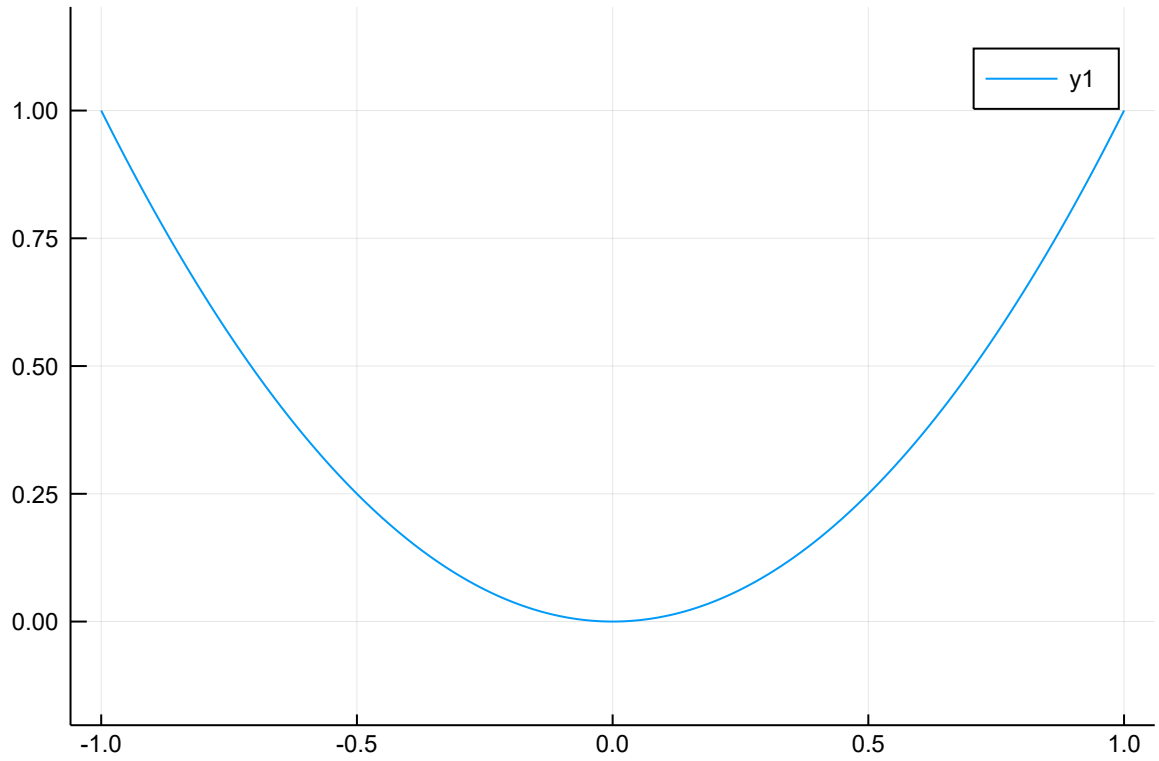
```
  [bac558e1] + OrderedCollections v1.1.0
  [ccf2f8ad] + PlotThemes v0.3.0
  [995b91a9] + PlotUtils v0.5.8
  [91a5bcdd] + Plots v0.26.3
  [3cdcf5f2] + RecipesBase v0.7.0
  [189a3867] + Reexport v0.2.0
  [ae029012] + Requires v0.5.2
  [992d4aef] + Showoff v0.3.1
  [a2af1166] + SortingAlgorithms v0.3.1
  [90137ffa] + StaticArrays v0.11.0
  [2913bbd2] + StatsBase v0.32.0
  Building GR ———→ `C:\Users\osval\.julia\packages\GR\ZI5OE\deps\build.log`
  Building FFMPEG → `C:\Users\osval\.julia\packages\FFMPEG\bkWgb\deps\build.l
og`
  Building Plots —→ `C:\Users\osval\.julia\packages\Plots\h3o4c\deps\build.lo
g`
```

In [91]:
```
using Plots; # using is the "import"
```

```
┌ Info: Precompiling Plots [91a5bcdd-55d7-5caf-9e0b-520d859cae80]
└ @ Base loading.jl:1242
```

In [92]:
```
x = [i for i = -1:0.01:1];
plot(x, x.^2,aspect_ratio=:equal)
```

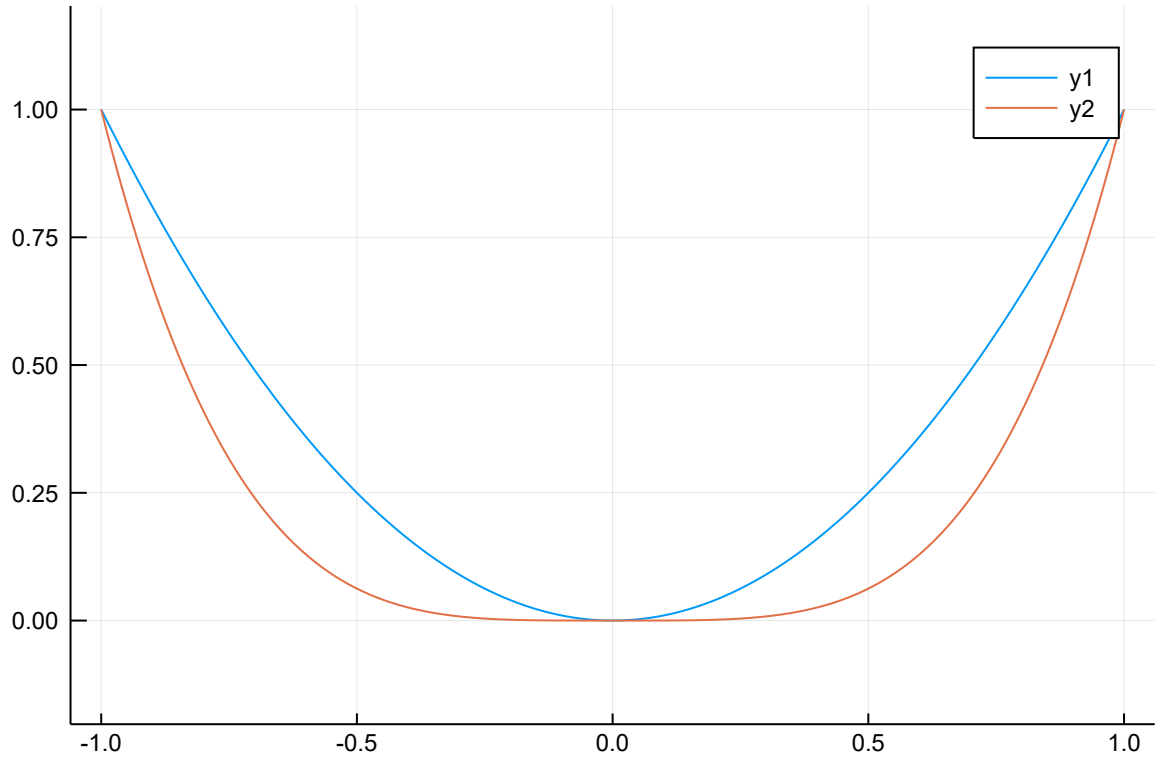
Out[92]:



By convention, ! indicates that a function will change the mutable object that it's acting on

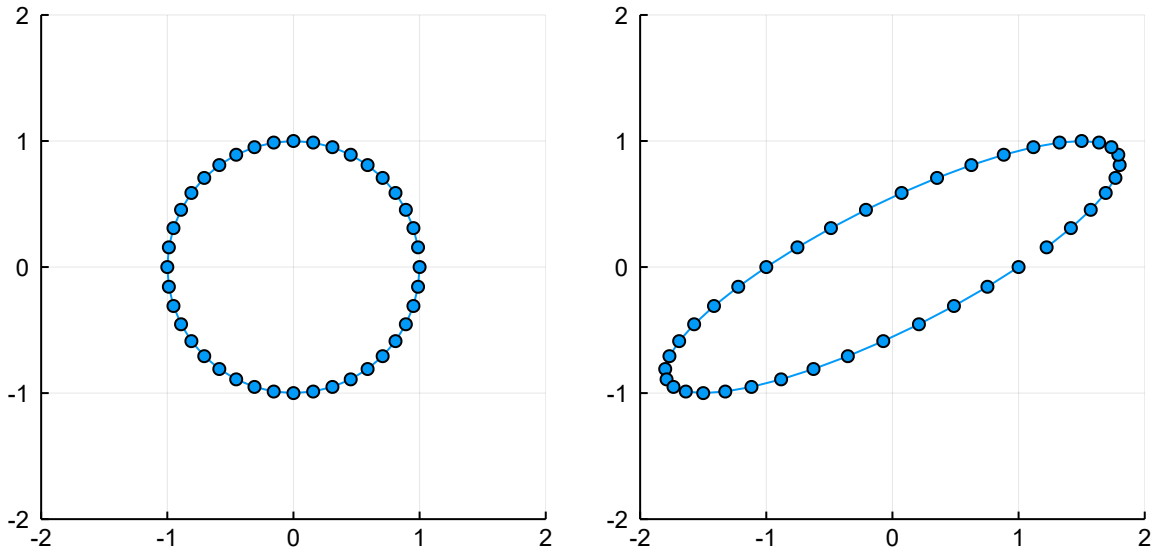
In [93]:
```
plot!(x, x.^4,aspect_ratio=:equal) # the "!" makes it modify the existing plot rather than replace it
```

Out[93]:



In [94]:
```
p1 = plot(1, xlim=(-2,2), ylim=(-2,2), marker = 3, aspect_ratio = :equal)
p2 = plot(1, xlim=(-2,2), ylim=(-2,2), marker = 3, aspect_ratio = :equal)

for k = 1:40

    θ = 2*pi*k/40
    λ = 1.5

    u = R(θ) * [1; 0]
    v = S(λ) * u

    push!(p1, u[1], u[2])
    push!(p2, v[1], v[2])

end

plot(p1,p2,layout=(1,2),legend=false)
```

Out[94]:



Notice that in the above code, functions "act" on objects and are not "properties" of the objects. I.e., rather than having `obj.method(a, b)` , we have `method(obj, a, b)` .

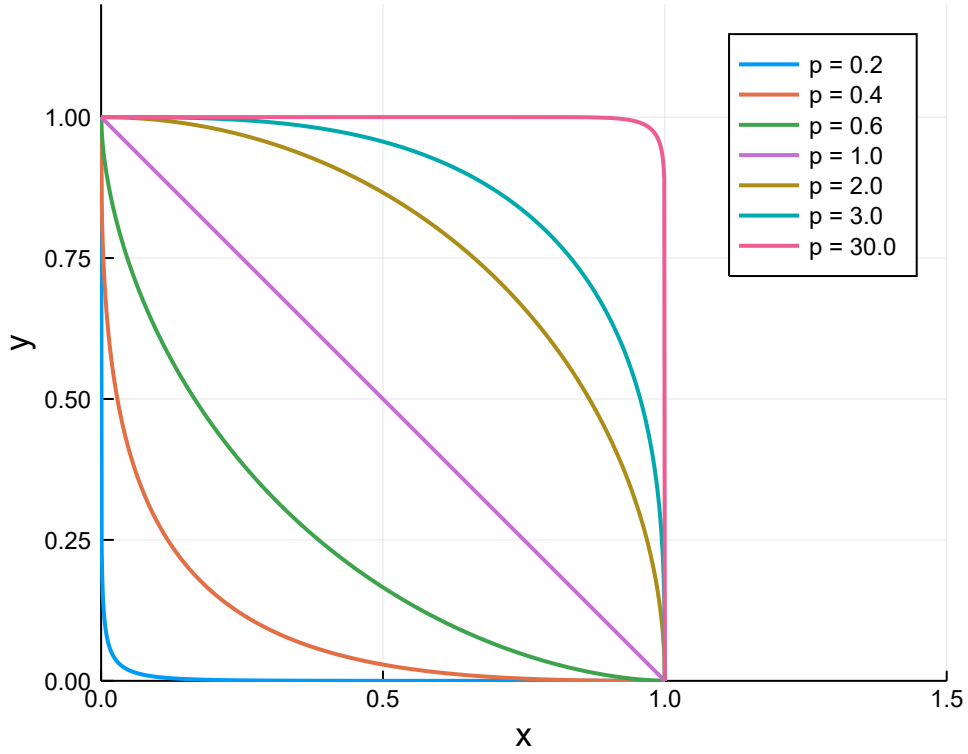
The L^p norm of a vector $\mathbf{x} \in \mathbf{R}^n$ is given by

$$||\mathbf{x}||_p = (x_1^p + x_2^p + \cdots + x_n^p)^{1/p}$$

Let $n = 2$. Exercise: Complete the the following code to plot the x, y pairs corresponding to $||\mathbf{x}||_p = 1$ for various p.

In [121]:
```
x = [i for i in LinRange(0,1,1000)];
p3 = plot()

for p = [0.2, 0.4, 0.6, 1, 2, 3, 30]

    y= (-(x.^p) .+ 1).^(1/p)

    plot!(p3, x, y, label = "p = $p")

end

plot(p3, xlim=(0,1.5), ylim=(0,1.2), aspect_ratio=:equal,
    size=(500,500),linewidth=2, xlabel="x", ylabel="y")
```

Out[121]:



In this problem you investigate how geometric concepts such as distance and angle can be applied to quantify similarity between text documents. You should have the files **`wordVecArticles.txt`**, **`wordVecTitles.txt`**, **`wordVecWords.txt`** and **`wordVecV.txt`** from the course website. The first two files each have ten lines. Each line in the first file consists of the text of one Wikipedia article. The corresponding line of the second file is the title of the article. The last two files are described in detail below.

Denote by D the set of documents where the number of documents is $|D|$. (In our dataset $|D| = 10$). Let W denote the union of words in all articles, i.e., the lexicon of the set of documents. We denote the cardinality of W by $|W|$. Assume the lexicon is ordered "lexiographically" (e.g., alphabetically) so that there is a one-to-one mapping from each word $w \in W$ to an element of the index set $t \in [|W|]$. Let $f_{\text{term}}(t, d)$ denote the number of times the word $w \in W$ that is indexed as $t \in [|W|]$ appears in the dth article where $d \in [|D|]$. Note that $\sum_{t=1}^{|W|} f_{\text{term}}(t, d)$ is the number of words (the length) of the dth article. We refer to $f_{\text{term}}(t, d)$ as the "term frequency" (really "term count").

A pre-computed W set and pre-computed $f_{\text{term}}(t, d)$ have been provided. The pre-processed data appears in the files **`wordVecWords.txt`** and **`wordVecV.txt`**. The first file represents the set W where elements of W are listed line by line, for 1651 lines, i.e., $|W| = 1651$. The file **`wordVecV.txt`** contains a matrix **`V`** of dimensions 1651×10. The value in the tth row and dth column of this matrix is $f_{\text{term}}(t, d)$. Use the provided data in **`V`** to answer parts (a) to (d) of this problem.

(a) Let the $|W|$-dimensional vectors v_d, $d \in [|D|]$ be defined as $v_d = (f_{\text{term}}(1, d), f_{\text{term}}(2, d), \ldots, f_{\text{term}}(|W|, d))$. Using v_d to represent the dth document, which two articles are closest in Euclidean distance (smallest distance)? Which two are closest in angle distance (smallest angle)? Are they the same pair, if not, what could be a reason for them being different? The functions `norm` and `findmin`, could be useful. Recall that you can read the documentation for a function using `?function`. You'll also need to compute pairwise distances. The Distances.jl package could be used (Google it). But you can also just do it by yourself, your code doesn't need to be efficient.

Some code for loading the files has been written to start you off.

In [122]:
```
file = open("wordVecTitles.txt")
```

Out[122]:
```
IOStream(<file wordVecTitles.txt>)
```

In [123]:
```
titles = readlines(file)
```

Out[123]:
```
10-element Array{String,1}:
 "B. J. Cole"
 "Mary J. Blige"
 "Jessica Feshbach"
 "Susie Au"
 "Geoff Brown (baseball)"
 "John Holland (composer)"
 "James Forder"
 "Public image of George W. Bush"
 "Barack Obama"
 "George W. Bush"
```

In [124]:
```
close(file)
```

In [125]:
```
file = open("wordVecWords.txt")
words = readlines(file)
close(file)
```

In [126]:
```
using DelimitedFiles
```

In [127]:
```
V = readdlm("wordVecV.txt", ',', Float64, '\n') # The entries of this are f_term(t,d)
```

Out[127]:
```
1651×10 Array{Float64,2}:
 0.0  0.0  0.0  0.0  1.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  1.0  0.0  1.0  0.0  0.0  0.0  0.0
 1.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  4.0  0.0  0.0  1.0  0.0  0.0  0.0  0.0  0.0
 0.0  2.0  0.0  0.0  0.0  0.0  0.0  1.0  0.0  2.0
 0.0  0.0  0.0  0.0  1.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0  1.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  1.0  0.0
 1.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  2.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  1.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
 1.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
 1.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
 ⋮                        ⋮
 0.0  1.0  0.0  0.0  0.0  1.0  0.0  0.0  1.0  0.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  1.0  0.0  0.0
 0.0  0.0  0.0  1.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  1.0  0.0  0.0
 0.0  0.0  0.0  1.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  1.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0  0.0  2.0  0.0  0.0  0.0  0.0
 0.0  1.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
 1.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  1.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  1.0  0.0  0.0  0.0  0.0  0.0  0.0
 1.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
```

In [143]:
```
function ED_min(V)
    lowest = Inf
    a=0
    b=0
    for i in 1:10
        for j in i+1:10
            if i != j
                ED = norm(V[:,i] - V[:,j])
                if ED < lowest
                    lowest = ED
                    a = i
                    b = j
                end
            end
        end
    end
    return (a,b)
end
```

Out[143]: ED_min (generic function with 1 method)

In [144]:
```
ED_min(V)
```

Out[144]: (7, 8)

In [155]:
```
function ang_min(V)
    lowest = Inf
    a=0
    b=0
    for i in 1:10
        for j in i+1:10
            if i != j
                ang = 1-(transpose(V[:,i])V[:,j])/(norm(V[:,i])*norm(V[:,j]))
                if  ang < lowest
                    lowest = ang
                    a = i
                    b = j
                end
            end
        end
    end
    return (a,b)
end
```

Out[155]: ang_min (generic function with 1 method)

In [156]:
```
ang_min(V)
```

Out[156]: (9, 10)

Type your answers to (a) in the Markdown cell below this one:

Answer: EuD min 7,8 cos min 9,10

They mightbe different becaues the euclidean distance is affected by article length

(b) In this part let the $|W|$-dimensional normalized vectors $\tilde{v}_d$, $d \in [|D|]$ be defined as $\tilde{v}_d = v_d / \sum_{t=1}^{|W|} f_{\text{term}}(t, d)$, where the v_d are defined as in the previous part. Using $\tilde{v}_d$ to represent the dth document, which two articles are closest in Euclidean distance (smallest distance)? Which two are closest in angle distance (smallest angle)? Are your answers the same as in the previous part? What would be a reason for using this normalization?

In [186]:
```
V_t = zeros(size(V))

for i in 1:10
    V_t[:,i] = V[:,i]./sum(V[:,i])
end
```

In [187]:
```
ED_min(V_t)
```

Out[187]: (9, 10)

In [188]:
```
ang_min(V_t)
```

Out[188]: (9, 10)

Type your answers to (b) in the Markdown cell below this one:

Answer: EuD min 9,10

cos min 9,10

answers are different than before. Normalization might be used so that distance is less affected by article length

Now, let $f_{\text{doc}}(t) = \sum_{d=1}^{|D|} \mathbb{I}[f_{\text{term}}(t,d) > 0]$ where $\mathbb{I}(\cdot)$ is the indicator function taking value one if the clause is true and zero else. The function $f_{\text{doc}}(t)$ counts in how many documents the tth word appears. We refer to $f_{\text{doc}}(t)$ as the document frequency.

We combine the term and document frequency definitions into what is called the term frequency-inverse document frequency score (TF-IDF), defined as

$$w(t,d) = \frac{f_{\text{term}}(t,d)}{\sum_{t=1}^{|W|} f_{\text{term}}(t,d)} \sqrt{\log\left(\frac{|D|}{f_{\text{doc}}(t)}\right)}.$$

Note, the denominator of the $\log$ is never zero since, by definition, each term appears in at least one document.

(c) Now let the $|W|$-dimensional vectors w_d, $d \in [|D|]$ be defined as $w_d = (w(1,d), w(2,d), \ldots, w(|W|,d))$. Using w_d to represent the dth document, which two articles are closest in Euclidean distance (smallest distance)? Which two are closest in angle distance (smallest angle)?

(d) What might be a reason for using the "inverse document frequency" adjustment? What is the adjustment doing geometrically?

In [238]:
```
#indicator = V .> 0
#size(indicator)
f_doc = zeros(size(V[:,1]))
for i in 1:1651
    f_doc[i] = sum(V[i,:].>0)
end

W = zeros(size(V))
for i in 1:1651
    for j in 1:10
        denom = sum(V[:,j])
        W[i,j] = V[i,j]/denom * (log(10 / f_doc[j])^0.5)
    end
end
size(W)
```

Out[238]: (1651, 10)

In [239]:
```
ED_min(W)
```

Out[239]: (9, 10)

In [240]:
```
ang_min(W)
```

Out[240]: (9, 10)

Type your answers to (c) and (d) in the Markdown cell below this one:

(c) Answer: EuD min 9,10

```
cos min 9,10
```

(d) to show how much information a word gives, geometrically it is scaling the vector so that words unique to few documents are weighted more heavily

OPTIONAL EXERCISE: (e) Write code to obtain W and $f_{\text{term}}(t, d)$ (which we named `V`) from the raw data files (articles and titles). Store the results in variables named `myW` and `myV`

In [ ]:
```
# Start here
```

In [ ]:
```
@assert myV == V
```

In [ ]:
```
@assert myW == words
```