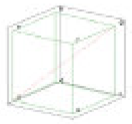


Table of Contents

ch01 sec1.1 几何视角下的线性方程组

二元一次方程的图像

二元一次方程组图像(两个方程)

三元元一次方程组

几何学处理不了四元一次方程组的问题.


```
begin
        using PlutoUI   , Plots   ,DataFrames   ,HypertextLiteral   ,LaTeXStrings
,Symbolics
        gr()
        theme(:bright)
        @htl("""<script src="https://cdn.bootcdn.net/ajax/libs/mathjax/3.2.0/es5/tex-svg-full.min.js"></script>
        """)
        PlutoUI.TableOfContents()
 end
```

ch01 sec1.1 几何视角下的线性方程组

Outcomes

方程组的解集类型, 两个变量的线性方程组考虑的是平面内直线的交集问题, 三个变量考虑的是三维空间中平面的交集问题

数学的威力在于对问题的泛化能力, 通过总结出数学对象共有的性质, 从而提出系统性解决问题的方法. 但是在学习数学概念的时候应该要从一般性跳出来, 寻找具体的实例来研究数学对象的性质. 这样做为我们理解抽象概念提供实例.

二维和三维空间下的线性方程组有一般线性方程组的性质, 同时也有自己的特性,就是可以使用几何方法直观化, 这为我们学习概念提供很好的工具. 可视化并不是目的, 可视化是帮助理解的工具.

数学对象就像一个实实在在的人一样客观存在, 有和其他人一样的共性, 也有自己的特性. 要充分理解一个数学对象 既要理解共性, 也要理解特性

```
md"""
 # ch01  sec1.1  几何视角下的线性方程组

!!! outcomes
      方程组的解集类型， 两个变量的线性方程组考虑的是平面内直线的交集问题， 三个变量考虑的是三维空间中平面的交集问题



数学的威力在于对问题的泛化能力， 通过总结出数学对象共有的性质， 从而提出系统性解决问题的方法． 但是在学习数学概念的时候应该要从一般性跳出来， 寻找具体的实例来研究数学对象的性质． 这样做为我们理解抽象概念提供实例．

二维和三维空间下的线性方程组有一般线性方程组的性质， 同时也有自己的特性，就是可以使用几何方法直观化， 这为我们学习概念提供很好的工具． 可视化并不是目的， 可视化是帮助理解的工具．


数学对象就像一个实实在在的人一样客观存在， 有和其他人一样的共性， 也有自己的特性． 要充分理解一个数学对象
既要理解共性， 也要理解特性\



"""
```

二元一次方程的图像

我们上初中时最先就接触了二元一次方程. 例如:

$$2x + 3y = 6$$

由于两个变量的组合是一个常数, 其中一个变量确定以后, 另一个变量也会确定下来. 第一个变量就称为自变量(independent), 第二个变量称为因变量(dependent). 经过移项和化简可以看到两者的关系:

$$y = -\frac{2}{3}x + 2$$

这是一个函数表达式, x 的取值位于定义域, y 属于值域, 表达式定义了两者之间的映射规则

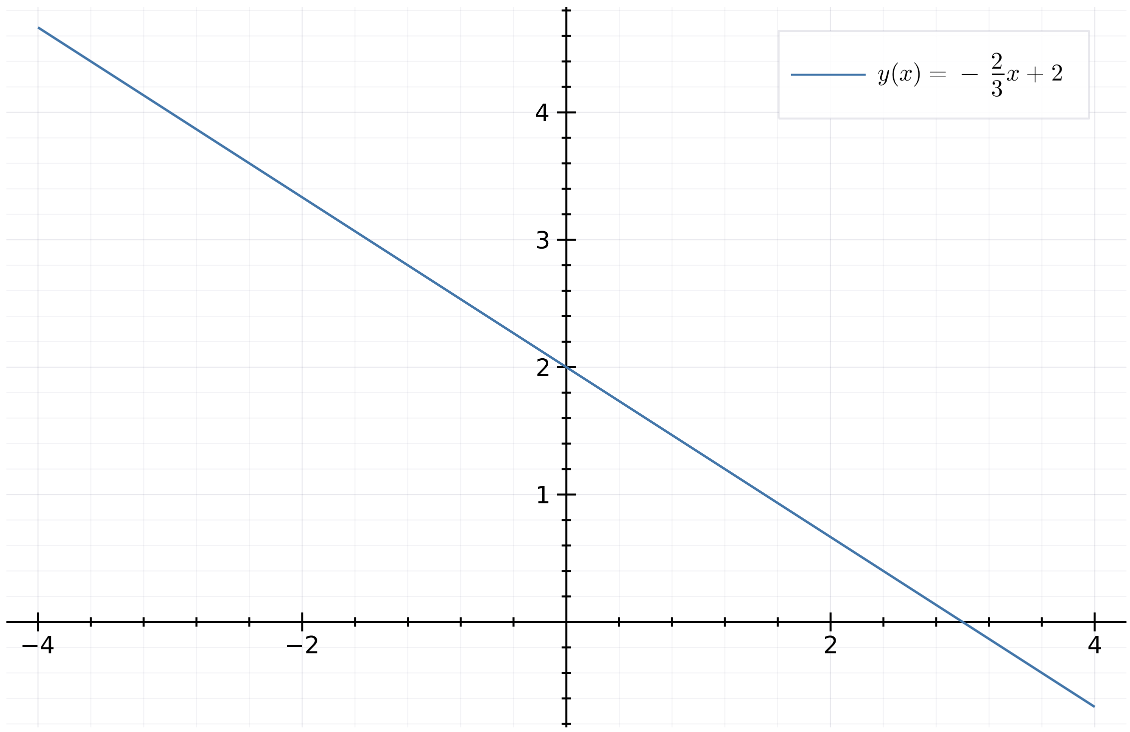
方程描述了两个变量要映射为一个实数的问题.

函数表达式刻画了自变量和因变量之间的关系.

Notice

理解函数的三个特性对于线性代数的学习是很重要的. 线性代数最核心的概念线性组合其实也是线性映射,根据矩阵里的信息, 把一个维度空间的量映射到另一个维度的空间. 后面我们会反复提到这个概念

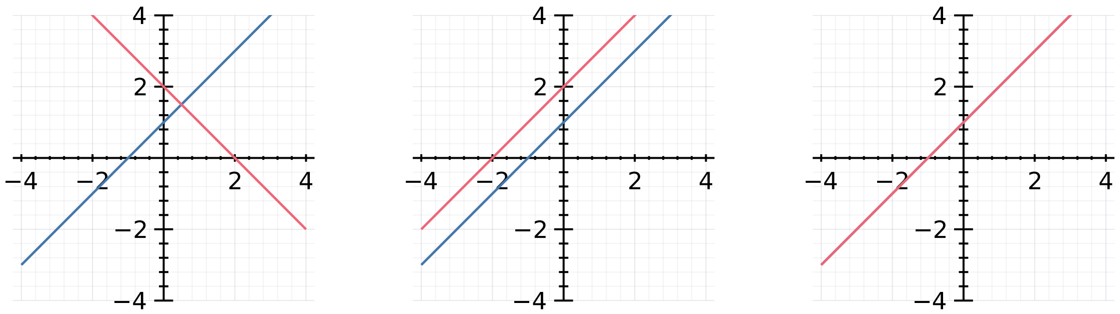
根据函数表达式可以画出两个变量之间的关系:



可以看到两个变量的关系是一条直线, 取一个x,在直线上可以找到对应的y 值.

二元一次方程组图像(两个方程)

平面内两条直线的关系有三种:1 相交, 2 平行, 3 重合:



1. 相交, 两条直线交于一点, 有一对(x, y) 既满足第一个方程, 又满足第二个方程
2. 平行, 两条直线没有交点, 没有(x, y) 同时在两条直线上
3. 重合,两条直线上所有的点同时满足方程1 和方程2

Example

example 1:用图形解方程组

$$x + y = 3$$

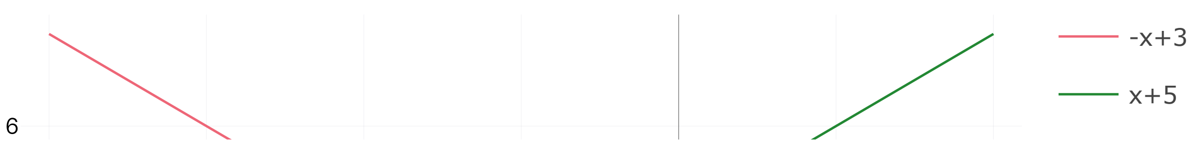
$$y - x = 5$$

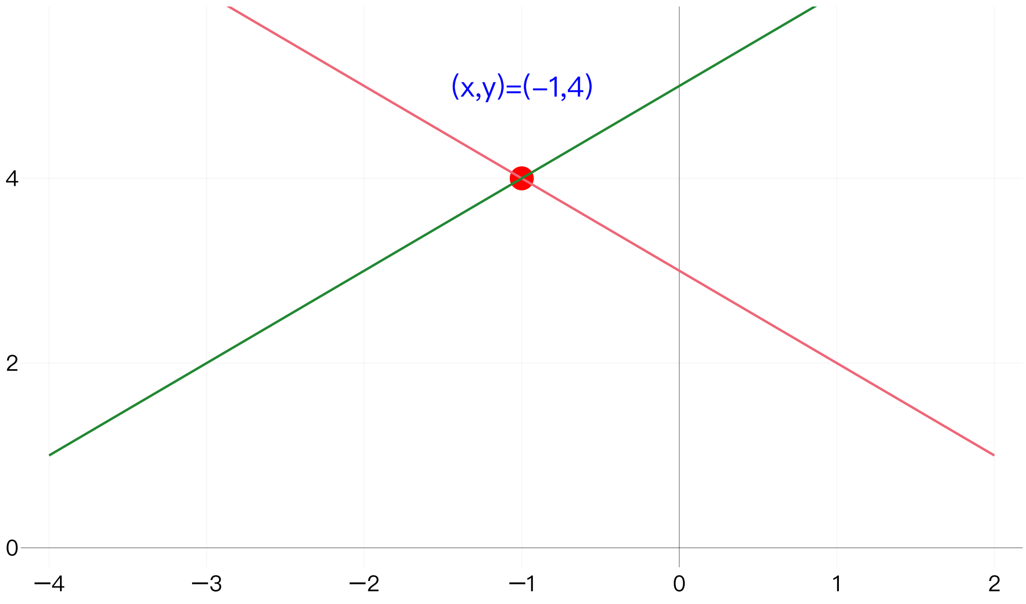
方程组经过化简可以得到:

$$y = -x + 3$$

$$y = x + 5$$

实际上就是看看这两条直线属于以上图中那种情况,图形如下:





可以之间看到方程组有一个解

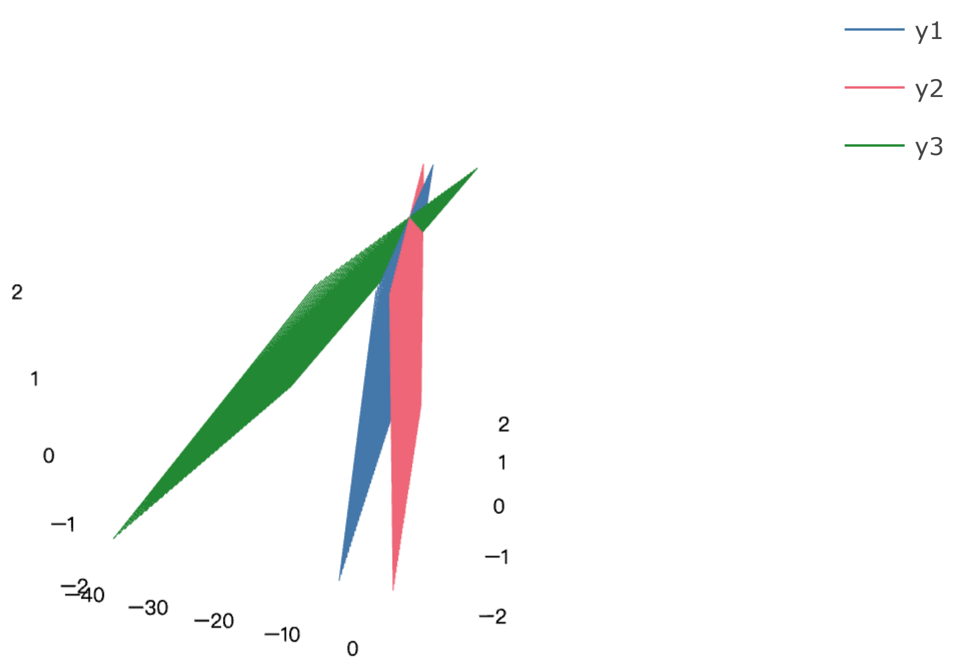
三元元一次方程组

如果是三元一次方程 , 例如$2x+4y-5z=8$

与二元一次方程类似, 这里三个变量决定一个常数, 当x, y 的值确定以后, z 的值就固定下来, x, y 是自变量, z 是因变量, 方程组可以表示为z 的函数,是一个平面 化简为:

$$z(x,y)=\frac{2x+4y-8}{5}$$

如果有方程组有三个方程, 一般形式的图像为:



这里展示的是三个平面相交的情况, 还有几种情况不再继续讨论

几何学处理不了四元一次方程组的问题.

我们可以列出四元一次方程的方程式:

$$x + y - 2z + 4w = 8$$

但是没有办法对其进行可视化, 三个自变量已经把三维空间填满,因变量没有生存的空间.

这个时候需要新的方法和视角. 其实从三维空间到四维空间的过渡历经了几百年时,这个过渡借助的是代数方法.

下一节我们就把视点转向代数

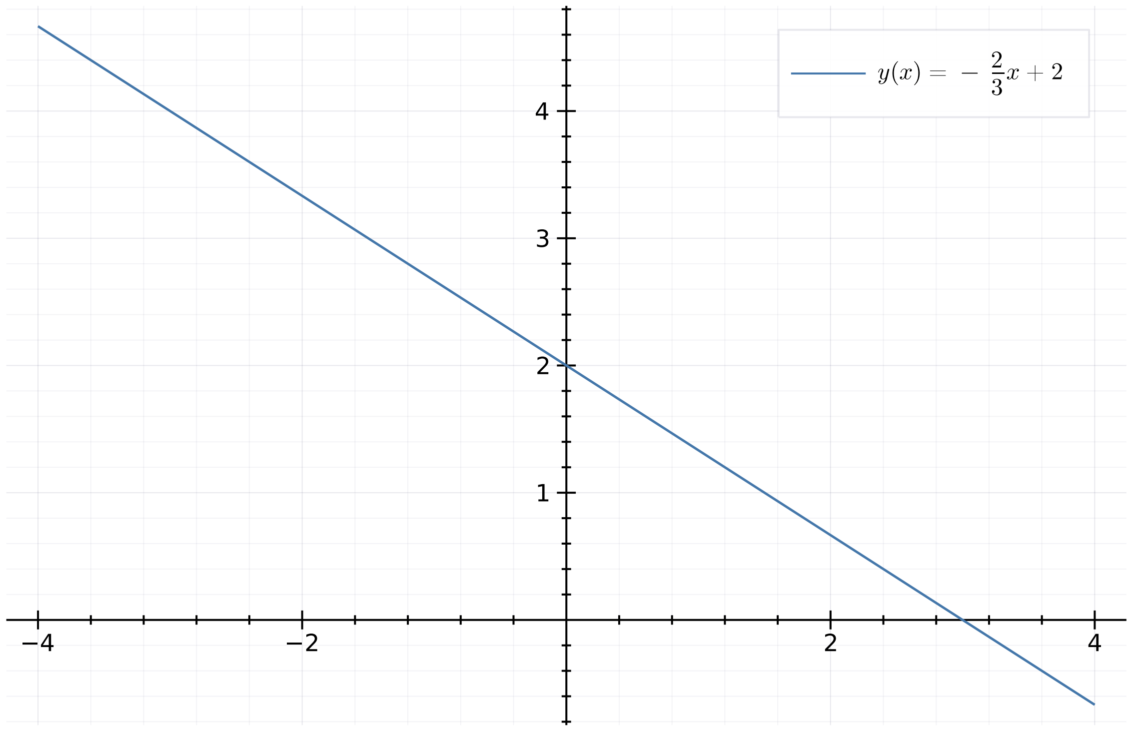
但是在线性代数的学习中药注意一点, 我们并不局限于某种视点, 在面对不同的问题时切换到容易使用的方法上是最有效的办法.

```
md"""
## 二元一次方程的图像

我们上初中时最先就接触了二元一次方程.
例如:

$2x+3y=6$

由于两个变量的组合是一个常数, 其中一个变量确定以后, 另一个变量也会确定下来. 第一个变量就称为自变量(independent), 第二个变量称为因变量(dependent). 经过移项和化简可以看到两者的关系:

$y=-\frac{2}{3}x+2$

这是一个函数表达式, $x$ 的取值位于定义域, $y$ 属于值域, 表达式定义了两者之间的映射规则

方程描述了两个变量要映射为一个实数的问题.

函数表达式刻画了自变量和因变量之间的关系.

!!! notice
    理解函数的三个特性对于线性代数的学习是很重要的. 线性代数最核心的概念线性组合其实也是线性映射,根据矩阵里的信息, 把一个维度空间的量映射到另一个维度的空间. 后面我们会反复提到这个概念


根据函数表达式可以画出两个变量之间的关系:

  $(store["l1"])

可以看到两个变量的关系是一条直线, 取一个$x$,在直线上可以找到对应的$y$ 值.


## 二元一次方程组图像(两个方程)

 平面内两条直线的关系有三种:1 相交, 2 平行, 3  重合:

```

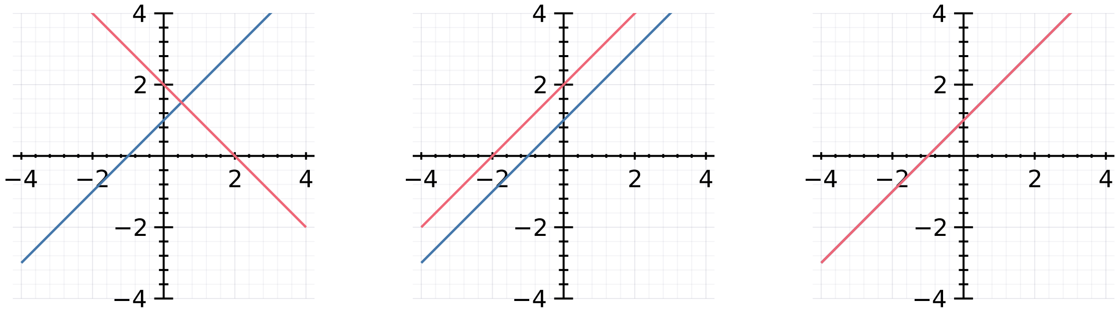
```
 $(store["2linetype"])


 1. 相交, 两条直线交于一点, 有一对$(x,y)$ 既满足第一个方程, 又满足第二个方程

 2. 平行, 两条直线没有交点, 没有$(x,y)$ 同时在两条直线上

 3. 重合,两条直线上所有的点同时满足方程$1$ 和方程$2$


!!! example
    example 1:用图形解方程组

    $x+y=3$

    $y-x=5$

方程组经过化简可以得到:

$y=-x+3$
$y= x+5$

实际上就是看看这两条直线属于以上图中那种情况,图形如下:

 $(store["2line"])

可以之间看到方程组有一个解



## 三元元一次方程组

 如果是三元一次方程 , 例如$2x+4y-5z=8$

与二元一次方程类似, 这里三个变量决定一个常数, 当$x,y$ 的值确定以后, $z$ 的值就固定下来, $x,y$ 是
自变量, $z$ 是因变量, 方程组可以表示为$z$ 的函数,是一个平面 化简为:

$z(x, y) =\frac{2x+4y-8}{5}$
如果有方程组有三个方程, 一般形式的图像为:

 $(store["3plane"])

这里展示的是三个平面相交的情况, 还有几种情况不再继续讨论


## 几何学处理不了四元一次方程组的问题.

我们可以列出四元一次方程的方程式:

$x+y-2z+4w=8$

但是没有办法对其进行可视化, 三个自变量已经把三维空间填满,因变量没有生存的空间.

这个时候需要新的方法和视角. 其实从三维空间到四维空间的过渡历经了几百年时,这个过渡借助的是代数方法.

下一节我们就把视点转向代数
```

但是在线性代数的学习中药注意一点，我们并不局限于某种视点，在面对不同的问题时切换到容易使用的方法上是最有效的办法.

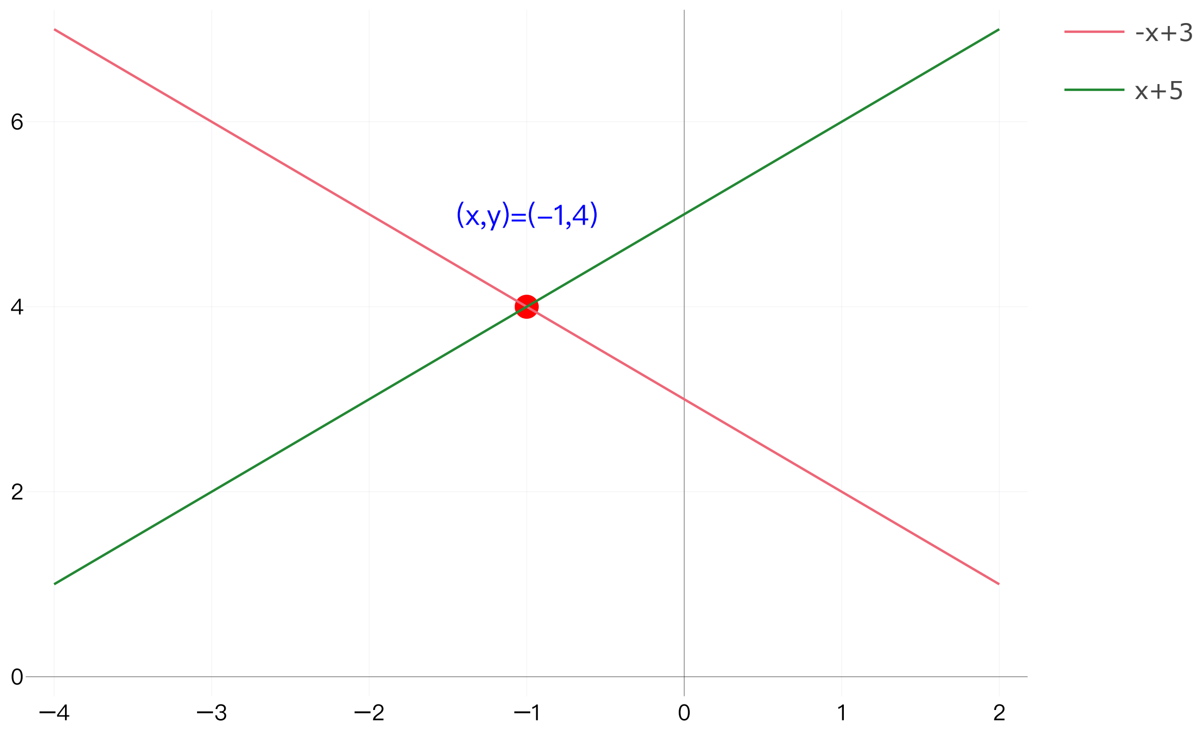
"""



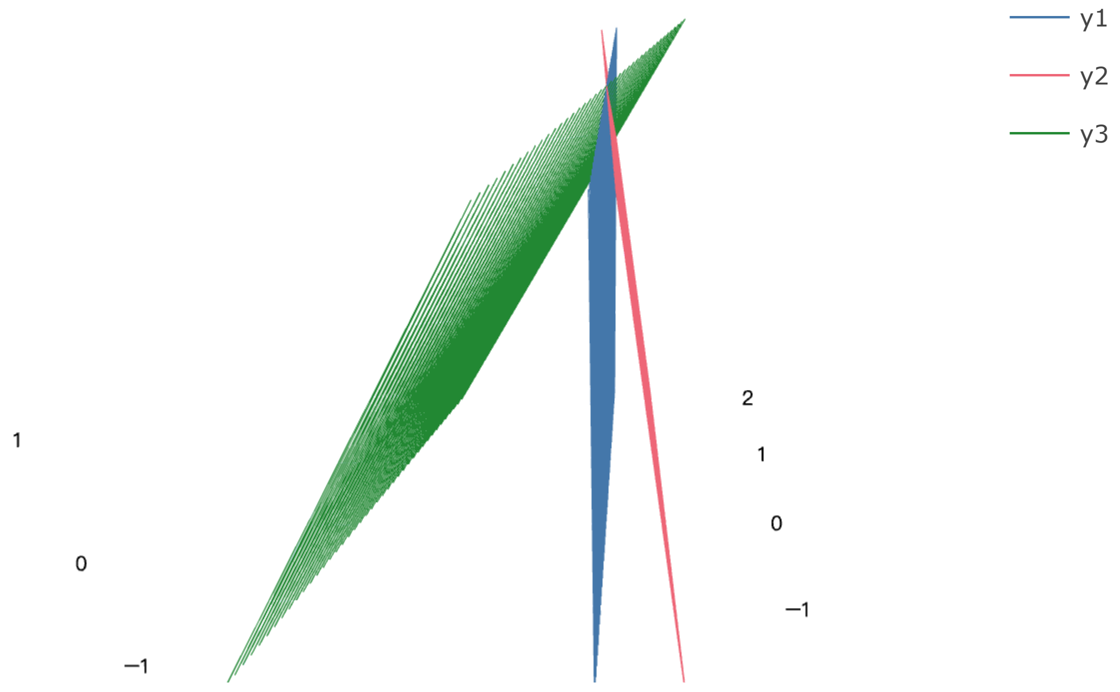

```
let

    tspan=-4:4
    f(x)=(-2/3)*x+2
    line1=plot(f, tspan, label=L"y(x)=-\frac{2}{3}x+2",frame=:origin)
    save("l1",line1)
end
```




```
let

    tspan=-4:4
    l1(x)=x+1
    l2(x)=x+2
    l3(x)=-x+2
    l4(x)=x+1
    p1=plot([l1,l3], tspan,frame=:origin,ratio=:equal,label=false,ylims=(-4,4))
    p2=plot([l1,l2], tspan, frame=:origin,ratio=:equal,label=false,ylims=(-4,4))
    p3=plot([l1,l4], tspan,frame=:origin,ratio=:equal,label=false,ylims=(-4,4))
    p4=plot!(p1,p2,p3,layout=(1,3))
    save("2linetype",p4)
end
```




```
let
    plotly()
    tspan=-4:2
    f1(x)=-x+3
    f2(x)=x+5
    ann=[
        (-1,5,text("(x,y)=(-1,4)",pointsize=10,color=:blue))
    ]
    scatter([-1],[4], ms=6, mc=:red,label=false)
    line2=plot!([f1,f2],tspan,frame=:origin,label=["-x+3" "x+5"],ann=ann)

    save("2line",line2)
end
```




```
let
    f1(x, y) =1/5*(2x+4y-8)
    f2(x, y) =-1/5*(2x+4y-8)
    f3(x, y) =2*(2x+4y-8)
    xs = collect(-2:0.1:2.0)
    ys = collect(-2:0.1:2.0)
    x_grid = [x for x = xs for y = ys]
    y_grid = [y for x = xs for y = ys]
    plane3=plot([f1.(x_grid,y_grid),f2.(x_grid,y_grid),f3.(x_grid,y_grid)],x_grid,
y_grid )

    save("3plane",plane3)
end
```

```
read (generic function with 1 method)
```

```
@htl("""<script src="https://cdn.bootcdn.net/ajax/libs/mathjax/3.2.0/es5/tex-svg-
full.min.js"></script>
""")
```