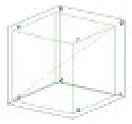


Table of Contents


```
begin
        using PlutoUI   , Plots   ,DataFrames   ,HypertextLiteral   ,LaTeXStrings   ,Symbolics   ,LinearAlgebra   ,RowEchelon
        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
```

ch02 sec2.6 点积

Outcomes

- A. 计算向量的点积
- B. 借助点积的性质, 包括柯西不等式, 三角不等式来证明向量相等或者不等
- C. 借助点积判断两个向量是否正交
- D. 计算一个向量在另一个向量上的投影
- E. 把向量分解为正交的组分

向量间的乘法有两种, 一种是点积, 一种是叉积, 这一小节先来看看点积

```
md"""
# ch02 sec2.6 点积

!!! outcomes
    - A. 计算向量的点积
    - B. 借助点积的性质, 包括柯西不等式, 三角不等式来证明向量相等或者不等
    - C. 借助点积判断两个向量是否正交
    - D. 计算一个向量在另一个向量上的投影
    - E. 把向量分解为正交的组分



向量间的乘法有两种, 一种是点积, 一种是叉积, 这一小节先来看看点积
"""
```

点积操作定义如下

Definition

给定R^n 中两个向量u: $\begin{bmatrix} u_1\\ \vdots \\u_n \end{bmatrix}$ 和 $v:\begin{bmatrix} v_1\\ \vdots \\v_n \end{bmatrix}$,点积定义为:

$$u \cdot v= u_1v_1+\cdots+u_nv_n$$

两个向量对应位置的分量乘积, 然后求和. 结果会得到一个标量.

```
md"""

点积操作定义如下

!!! definition

    给定$R^n$ 中两个向量u:$\begin{bmatrix} u_1\\ \vdots \\u_n \end{bmatrix}$ 和 $v:\begin{bmatrix} v_1\\ \vdots \\v_n \end{bmatrix}$,点积定义为:

    $u \cdot v= u_1v_1+\cdots+u_nv_n$

两个向量对应位置的分量乘积, 然后求和. 结果会得到一个标量.

"""
```

Example

example 1

计算 $u=[1,2,0,-1]'$ 和$v=[0,1,2,3]'$ 的点积

$$u\cdot v=(1)(0)+(2)(1)+(0)(2)+(-1)(3)$$

$$=0+2+0-3$$

$$=-1$$

```
md"""
!!! example

    example 1

    计算 $u=[1,2,0,-1]'$ 和$v=[0,1,2,3]'$ 的点积

$u\cdot v=(1)(0)+(2)(1)+(0)(2)+(-1)(3)$
$=0+2+0-3$
$=-1$
"""



```

-1

```julia
dot([1,2,0,-1],[0,1,2,3])  # julia  自带点积计算方法
```

点积的性质

Props

- $u \cdot v = v \cdot u$
- $u \cdot u \geq 0$, $u \cdot u = 0$ 只有在$u = 0$ 时满足条件
- $(ku + lv) \cdot w = k(u \cdot w) + l(v \cdot w)$
- $u(kv + lw) = k(u \cdot v) + l(u \cdot w)$
- $u \cdot u = ||u||^2$

```julia
md"""
## 点积的性质

!!! props

    - $u \cdot v=v \cdot u$
    - $u\cdot u \geq 0$, $u\cdot u=0$ 只有在$u=0$ 时满足条件
    - $(ku+lv)\cdot w=k(u\cdot w)+l(v\cdot w)$
    - $u(kv+lw)=k(u\cdot v)+l(u\cdot w)$
    - $u\cdot u=||u||^2$
"""
```

Example

example 2 用点积计算向量的长度

给定向量 $u=\begin{bmatrix}2\\ 1\\4 \\2 \end{bmatrix}$,使用点积计算长度

根据点积的性质可以得到 $u\cdot u=||u||^2$ 所以有:

$$|u|=\sqrt{u\cdot u}$$

$$u\cdot u= 2^2+1^2+4^2+2^2=25$$

所以 $|u|=5$

```
md"""

!!! example
    example 2  用点积计算向量的长度

    给定向量 $u=\begin{bmatrix}2\\ 1\\4 \\2 \end{bmatrix}$,使用点积计算长度


根据点积的性质可以得到
$u\cdot u=||u||^2$
所以有:

$|u|=\sqrt{u\cdot u}$

$u\cdot u= 2^2+1^2+4^2+2^2=25$
所以 $|u|=5$
"""
```

```
5.0
```

```
let
    u=[2,1,4,2]
    sqrt(dot(u,u))
end
```

柯西不等式和三角不等式

柯西不等式由点积衍生而来, 定义为:

Definition

柯西不等式:

给定两个向量:u 和v 有下面不等式成立:

$$|u \cdot v| \leq ||u|| \ ||v||$$

如果两个向量方向相同, 则等式成立.

因此, 柯西不等式为判断两个向量的方向提供了数值依据

在柯西不等式基础上又衍生出了三角不等式

Definition

三角不等式:

给定两个向量:u 和v 有下面不等式成立:

$$||u+v|| \leq ||u|| + ||v||$$

两个向量与向量的和组成三角形, 两边之和大于等于第三边, 当两个向量方向相同时,等式成立

```
md"""
## 柯西不等式和三角不等式

柯西不等式由点积衍生而来，定义为:

!!! definition

    柯西不等式:

    给定两个向量:$u$ 和$v$  有下面不等式成立:

    $|u\cdot v| \leq ||u|| \ ||v||$

    如果两个向量方向相同，则等式成立.

    因此，柯西不等式为判断两个向量的方向提供了数值依据


在柯西不等式基础上又衍生出了三角不等式

!!! definition

    三角不等式:

```

```
    给定两个向量:$u$ 和$v$  有下面不等式成立:

    $||u+v|| \leq ||u|| + ||v||$

    两个向量与向量的和组成三角形, 两边之和大于等于第三边, 当两个向量方向相同时,等式成立
"""
```

点积的几何特性

两个向量的夹角定义为θ,范围在$0 \leq \theta \leq \pi$

使用点积可以求解两个向量间的夹角.

Definition

点积和向量夹角的关系

给定两个向量:u 和v 有下面的关系:

$$u \cdot v=||u|| \ ||v||cos\theta$$

```
md"""
## 点积的几何特性

两个向量的夹角定义为$\theta$,范围在$0 \leq \theta \leq \pi$

使用点积可以求解两个向量间的夹角.

!!! definition
    点积和向量夹角的关系

    给定两个向量:$u$ 和$v$  有下面的关系:

    $u \cdot v=||u|| \ ||v||cos\theta$
"""
```

Example

example 3

求向量 $\begin{bmatrix}2\\2\end{bmatrix},\begin{bmatrix}0\\3\end{bmatrix}$ 之间的夹角.

根据夹角公式变形得到:

$$cos\theta=\frac{u \cdot v}{||u|| \ ||v||}$$

$$0.25\pi$$

```
let
    vec1,vec2=[2,2],[0,3]
    inc_angle(vec1,vec2)
end
```

正交向量

正交的意思是两个向量的夹角为90° . 在直角坐标系中, 我们一直都在使用正交向量, 二维坐标系下的分别与x , y 轴方向相同的向量之间都是正交的. 在几何里称为垂直.

在线性代数中约定任何向量与零向量都正交. 这个约定没有几何意义, 只是为了理论完整性需要所设定.

Defintion

正交向量:

给定两个向量:u 和v 有下面的关系:

$$u \cdot v=0$$

我们就说 向量 $u \perp v$, 表示 u 和v 正交

因为有$cos90^{\circ}=0$, 所以余弦值为 0, 两个向量的正交

```
md"""
## 正交向量

正交的意思是两个向量的夹角为$90^{\circ}$ . 在直角坐标系中, 我们一直都在使用正交向量, 二维坐标系下的分别与$x \ , y$ 轴方向相同的向量之间都是正交的. 在几何里称为垂直.

在线性代数中约定任何向量与零向量都正交. 这个约定没有几何意义, 只是为了理论完整性需要所设定.


!!! defintion

    正交向量:

    给定两个向量:$u$ 和$v$  有下面的关系:

    $u \cdot v=0$

    我们就说 向量 $u \perp v$, 表示 $u$ 和$v$ 正交

   因为有$cos90^{\circ}=0$, 所以余弦值为 0, 两个向量的正交

"""
```

Example

example 4

判断下面两个向量是否正交, $u=\begin{bmatrix}2\\ 1\\ -1 \end{bmatrix}$,$v=\begin{bmatrix}1\\3 \\5 \end{bmatrix}$

从下面计算结果可以知道两个向量夹角为90°, 是正交的

```
md"""

!!! example

    example 4

    判断下面两个向量是否正交,
    $u=\begin{bmatrix}2\\ 1\\ -1 \end{bmatrix}$ ,$v=\begin{bmatrix}1\\3 \\5
\end{bmatrix}$

从下面计算结果可以知道两个向量夹角为$90^{\circ}$, 是正交的
"""
```

$$0.5\pi$$

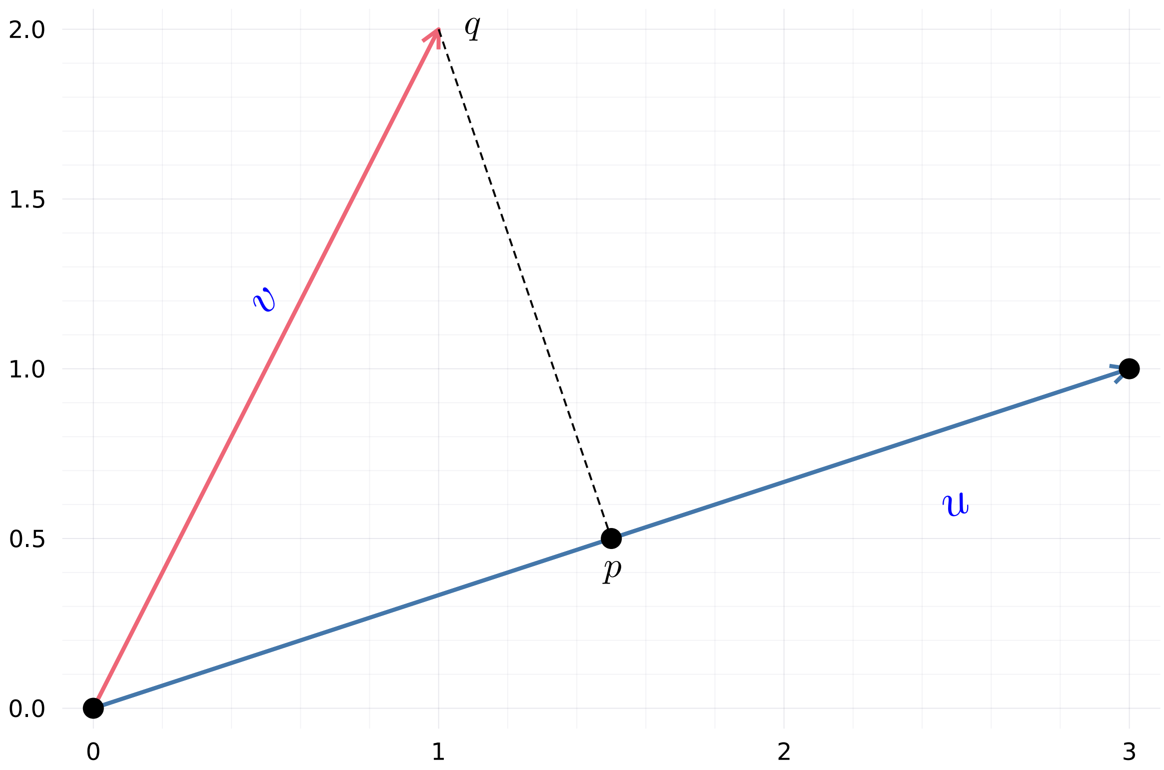
```
let
    u, v = [2,1,-1] , [1,3,5]
    inc_angle(u,v)
end
```

投影

计算一个向量与另一个向量方向相同的组分有很多应用, 获取一个向量在另一个向量方向上的组分就定义为投影.

定义二维空间向量 $u = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$,$v = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$

如图



向量 $\vec{oq}$ 在向量 $\vec{u}$ 方向上的投影表示为$\vec{op}$

向量 $\vec{u}$ 在向量 $\vec{v}$ 之间的夹角定义为θ

pq 与向量$\vec{u}$ 垂直.

o, p, q 组成一个直角三角形.

我们需要知道$\vec{op}$ 的坐标值.

首先在直角三角形中计算出$|op|$ 的长度

$$cos\theta = \frac{|op|}{|oq|} = \frac{|op|}{||v||}$$

所以有:

$$|op| = ||v||cos\theta$$

通过点积公式, 我们知道u, v 之间的关系为:

$$u \cdot v = ||u||\ ||v||cos\theta$$

化简后得到:

$$||v||cos\theta = \frac{u \cdot v}{||u||}$$

因此$|op|$ 表示为:

$$|op| = \frac{u \cdot v}{||u||}$$

上述计算结果定义为$\vec{v}$ 在$\vec{u}$ 方向的分量(不是向量)

$\vec{v}$ 在$\vec{u}$ 方向的分量乘以标准化的 $\vec{u}$ 向量就得到$\vec{v}$ 在$\vec{u}$ 上的投影

$$proj_u(v) = \vec{op} = \frac{|op|}{||u||}u = \frac{u \cdot v}{||u||^2}u$$

$$= \frac{u \cdot v}{u \cdot u}u$$

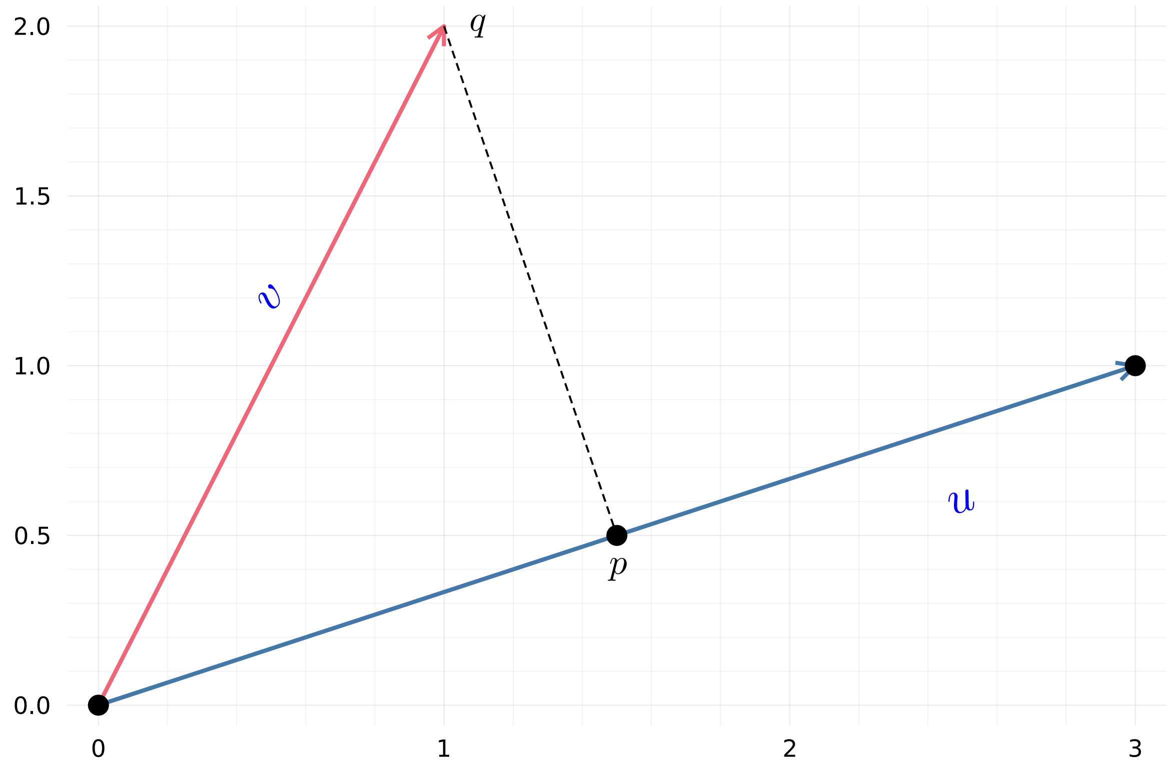
图中的p 点坐标就是根据这个公式计算得来

```
projvonu(v,u)=(dot(u,v)/dot(u,u))*u
```

```
md"""
## 投影

计算一个向量与另一个向量方向相同的组分有很多应用, 获取一个向量在另一个向量方向上的组分就定义为投影.

定义二维空间向量 $u=\begin{bmatrix}1\\ 2 \end{bmatrix}$
,$v=\begin{bmatrix}3\\1\end{bmatrix}$

如图

 $(store["proj"])

 向量 $\vec{oq}$ 在向量 $\vec{u}$ 方向上的投影表示为$\vec{op}$

  向量 $\vec{u}$ 在向量 $\vec{v}$ 之间的夹角定义为$\theta$

  $pq$ 与向量$\vec{u}$ 垂直.

  $o,p,q$ 组成一个直角三角形.

 我们需要知道$\vec{op}$ 的坐标值.

首先在直角三角形中计算出$|op|$ 的长度

$cos \theta =\frac{|op|}{|oq|}=\frac{|op|}{||v||}$

```

- 所以有:
-
- $|op|=||v||cos \theta$
-
- 通过点积公式, 我们知道u,v 之间的关系为:
-
- $u\cdot v=||u|| \ ||v||cos \theta$
-
- 化简后得到:
-
- $||v||cos \theta=\frac{u\cdot v}{||u||}$
-
- 因此$|op|$ 表示为:
-
- $|op|=\frac{u\cdot v}{||u||}$
-
- 上述计算结果定义为$\vec{v}$ 在$\vec{u}$ 方向的分量(不是向量)
-
- $\vec{v}$ 在$\vec{u}$ 方向的分量乘以标准化的 $\vec{u}$ 向量就得到$\vec{v}$ 在$\vec{u}$ 上的投影
-
- $proj_u(v)=\vec{op} =\frac{|op|}{||u||}u=\frac{u\cdot v}{||u||^2}u$
- $=\frac{u\cdot v}{u\cdot u}u$
-
- 图中的p 点坐标就是根据这个公式计算得来
-

```julia
   projvonu(v,u)=(dot(u,v)/dot(u,u))*u
```
"""




```
let
    zero,v,u=[0,0],[1,2],[3,1]
    proj=projvonu(v,u)
    ann=[
        (0.5,1.2,text(L"v",pointsize=16, color=:blue,rotation=45)),
        (2.5,0.6,text(L"u",pointsize=16, color=:blue,rotation=15)),
        (1.5,0.4,text(L"p",pointsize=12, color=:black,rotation=0)),
        (1.1,2.0,text(L"q",pointsize=12, color=:black,rotation=0)),
    ]
    p1=vec_plot(zero, u)
    p2=vec_plot!(zero, v)
    p3=scatter!([zero[1],u[1],proj[1]],
[zero[2],u[2],proj[2]],ms=6,mc=:black,label=false)
    p4=plot!([v[1],proj[1]],[v[2],proj[2]],ls=:dash,lw=1,color=:black,label=false)
    p5=plot!(ann=ann)
    save("proj",p5)

end
```

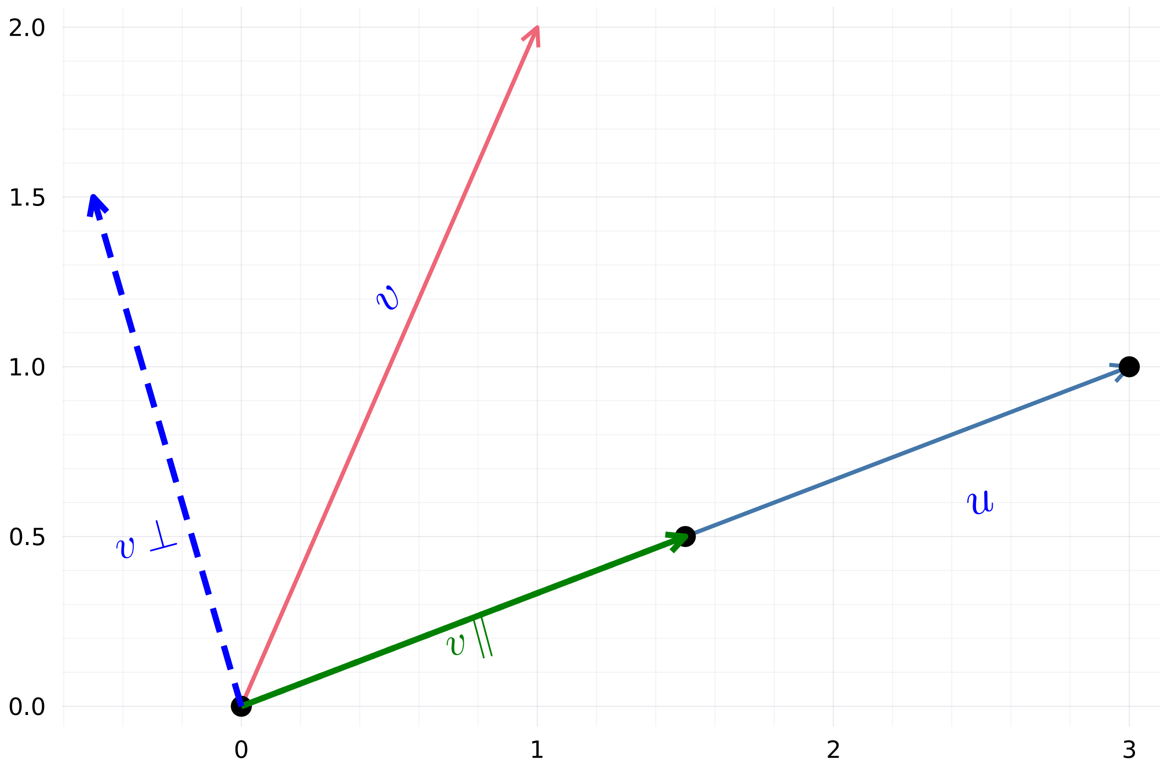
分解向量

Definition

假设一个非0 向量u, 给定一个向量v, 只存在两个向量,满足如下关系:

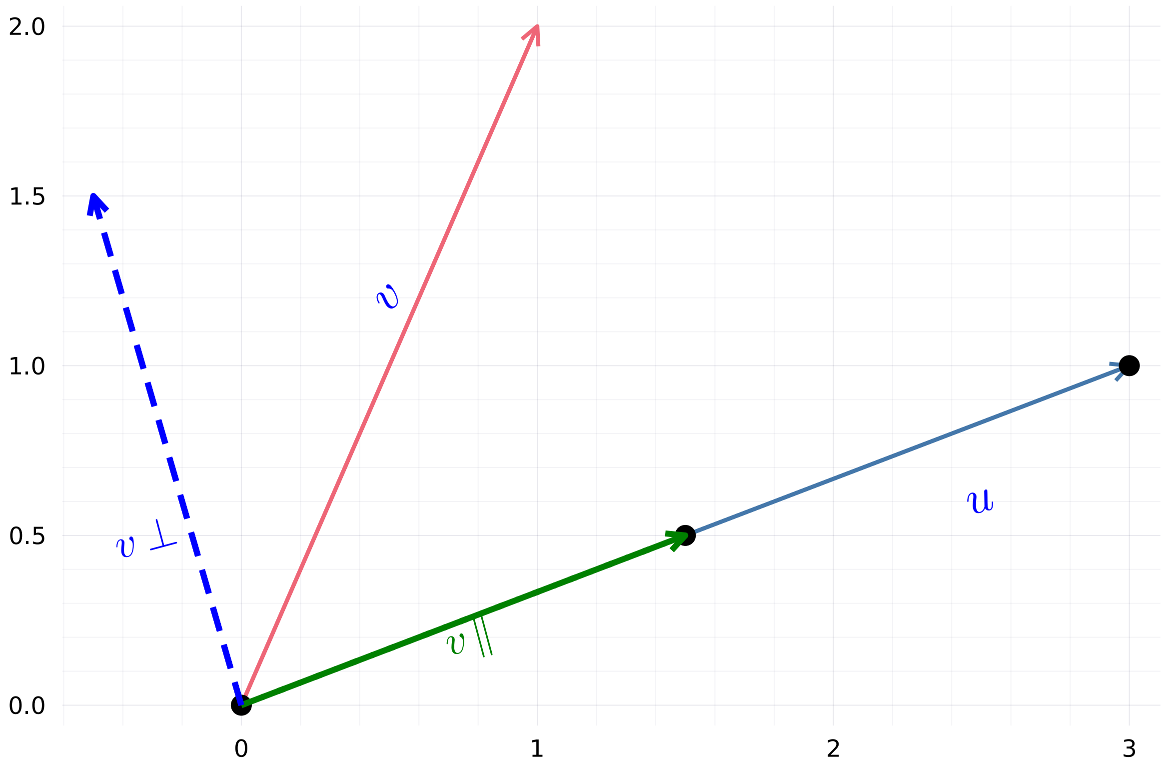
$v=v\parallel +v \perp$

$v\parallel$ 与u 有标量倍乘关系, $v \perp$ 正交于u



之前我们已经看过平行于u 的向量. 垂直于u 的向量为$v \perp=v-v\parallel$

```
md"""
## 分解向量

!!! definition

    假设一个非$0$ 向量$u$, 给定一个向量$v$, 只存在两个向量,满足如下关系:

     $v=v\parallel +v \perp$

     $v\parallel$ 与$u$ 有标量倍乘关系, $v \perp$ 正交于$u$

 $(store["decomposition"])

之前我们已经看过平行于$u$ 的向量.  垂直于$u$ 的向量为$v \perp=v-v\parallel$
"""
```




```julia
let
    zero,v,u=[0,0],[1,2],[3,1]
    proj=projvonu(v,u)
    perp=v-proj
    ann=[
        (0.5,1.2,text(L"v",pointsize=16, color=:blue,rotation=45)),
        (2.5,0.6,text(L"u",pointsize=16, color=:blue,rotation=15)),
        (0.8,0.2,text(L"v \parallel ",pointsize=14, color=:green,rotation=15)),
        (-0.3,0.5,text(L"v \perp ",pointsize=14, color=:blue,rotation=15))


    ]
    p1=vec_plot(zero, u)
    p2=vec_plot!(zero, v)
    p3=scatter!([zero[1],u[1],proj[1]],
[zero[2],u[2],proj[2]],ms=6,mc=:black,label=false)
    p4=plot!([zero[1],proj[1]],
[zero[2],proj[2]],ls=:dash,lw=3,color=:green,label=false,arrow=true)
    p5=plot!([zero[1],perp[1]],
[zero[2],perp[2]],ls=:dash,lw=3,color=:blue,label=false,arrow=true)
    p6=plot!(ann=ann)
    save("decomposition",p6)

end
```

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

```
begin
    function   inc_angle(vec1,vec2)
         vdot=dot(vec1,vec2)
         l1,l2=length(vec1),length(vec2)
         angle=round(acos(vdot/(l1*l2))/pi,digits=2)
         return L"%$(angle) \pi"

    end
     function   inc_angle2(vec1,vec2)
         vdot=dot(vec1,vec2)
         l1,l2=length(vec1),length(vec2)
         angle=round(acos(vdot/(l1*l2))/pi,digits=2)
         return angle

    end

    function length(vec)
             length=size(vec)
             arr=[(vec[i])^2 for i in 1:length[1]]
             return sqrt(sum(arr))
    end

    function projvonu(v,u)
           return (dot(u,v)/dot(u,u))*u
    end
end
```

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

```
begin
        store=Dict()

        function save(key::String, dict)
            store[key]=dict
        end

        function read(key::String)
            return  store[key]
        end


        function vec_plot(v1,v2,ls=:solid)
            v11,v12=v1[1],v1[2]
            v21,v22=v2[1],v2[2]
            return plot([v11,v21],[v12,v22],label=false, arrow=true, lw=2,ls=ls)
        end

        function vec_plot!(v1,v2,ls=:solid)
            v11,v12=v1[1],v1[2]
            v21,v22=v2[1],v2[2]
            return plot!([v11,v21],[v12,v22],label=false, arrow=true, lw=2,ls=ls)
        end

        function vec_plot3d(v1,v2,ls=:solid)
            v11,v12,v13=v1[1],v1[2],v1[3]
            v21,v22,v23=v2[1],v2[2],v2[3]
            return plot([v11,v21],[v12,v22],[v13,v23],label=false, lw=1,ls=ls)
        end

        function vec_plot3d!(v1,v2,ls=:solid)
            v11,v12,v13=v1[1],v1[2],v1[3]
            v21,v22,v23=v2[1],v2[2],v2[3]
            return plot!([v11,v21],[v12,v22],[v13,v23],label=false, lw=1,ls=ls)
        end

        function dist(p,q)
            length=size(p)
            arr=[(abs(p[i]-q[i]))^2 for i in 1:length[1]]
            return sqrt(sum(arr))
        end


end
```