

```
c      +-----+
c      + subroutine choose_box : select obs for a box +
c      +-----+
c      subroutine choose_box(nbox,idim,jdim,iext,jext,
+          stnlis,latlis,lonlis,vallis,nrstn,nstn,
+          lat,lon,dlat,dlon,rmax,nmax,nmin,
+          stnna,nrana,latna,lonna,valna,nana)
c      implicit none
c      ----- input and output variables -----
c      integer nbox,idim,jdim,iext,jext
c      integer nstn,nana,nmax,nmin
c      integer stnlis(nstn),stnna(nmax),nrstn(nstn),nrana(nmax)
c      real latlis(nstn),lonlis(nstn),latna(nmax),lonna(nmax)
c      real vallis(nstn),valna(nmax),lat,lon,rmax,dlat,dlon
c      ----- work variables -----
c      real slat,nlat,wlon,elon
c      integer istn
c 1  ----- select an area (iext*dlon or jext*dlat extended at each bdy) -
-
      if(nbox.eq.1) then
          slat=lat-((jdim/2)+jext)*dlat
          nlat=lat+((jdim/2)+jext)*dlat
          wlon=lon-((idim/2)+iext)*dlon
          elon=lon+((idim/2)+iext)*dlon
      else if(nbox.eq.9) then ! hard-coded
          slat=lat-7*dlat
          nlat=lat+7*dlat
          wlon=lon-7*dlon
          elon=lon+7*dlon
      else
          write(6,*)'nbox:',nbox,' is not defined in choose_box.f'
          stop
      endif
c 2  ----- select obs for the box -----
      nana=0
      do istn=1,nstn
          if(vallis(istn).ge.0.0 .and.
+          latlis(istn).ge.slat.and.latlis(istn).le.nlat.and.
+          lonlis(istn).ge.wlon.and.lonlis(istn).le.elon) then
              nana=nana+1
              if(nana.gt.nmax) then
                  nana=nana-1
                  write(6,*) ' nobs more than nmax for box :',lat,lon
                  return
              endif
              stnna(nana)=stnlis(istn)
              latna(nana)=latlis(istn)
              lonna(nana)=lonlis(istn)
              valna(nana)=vallis(istn)
              nrana(nana)=nrstn(istn)
          endif
      enddo
      return
end
```