Supporting Information

for

$\delta^{18}O$ as a tracer of PO_4^{3-} losses from agricultural landscapes

by
Naomi S. Wells[1*], Daren C. Gooddy[2], Mustefa Yasin Reshid[1], Peter J. Williams[2], Andrew C. Smith[3], Bradley D. Eyre[1]

submitted to
Journal of Environmental Management

Table of contents

S1: Site maps

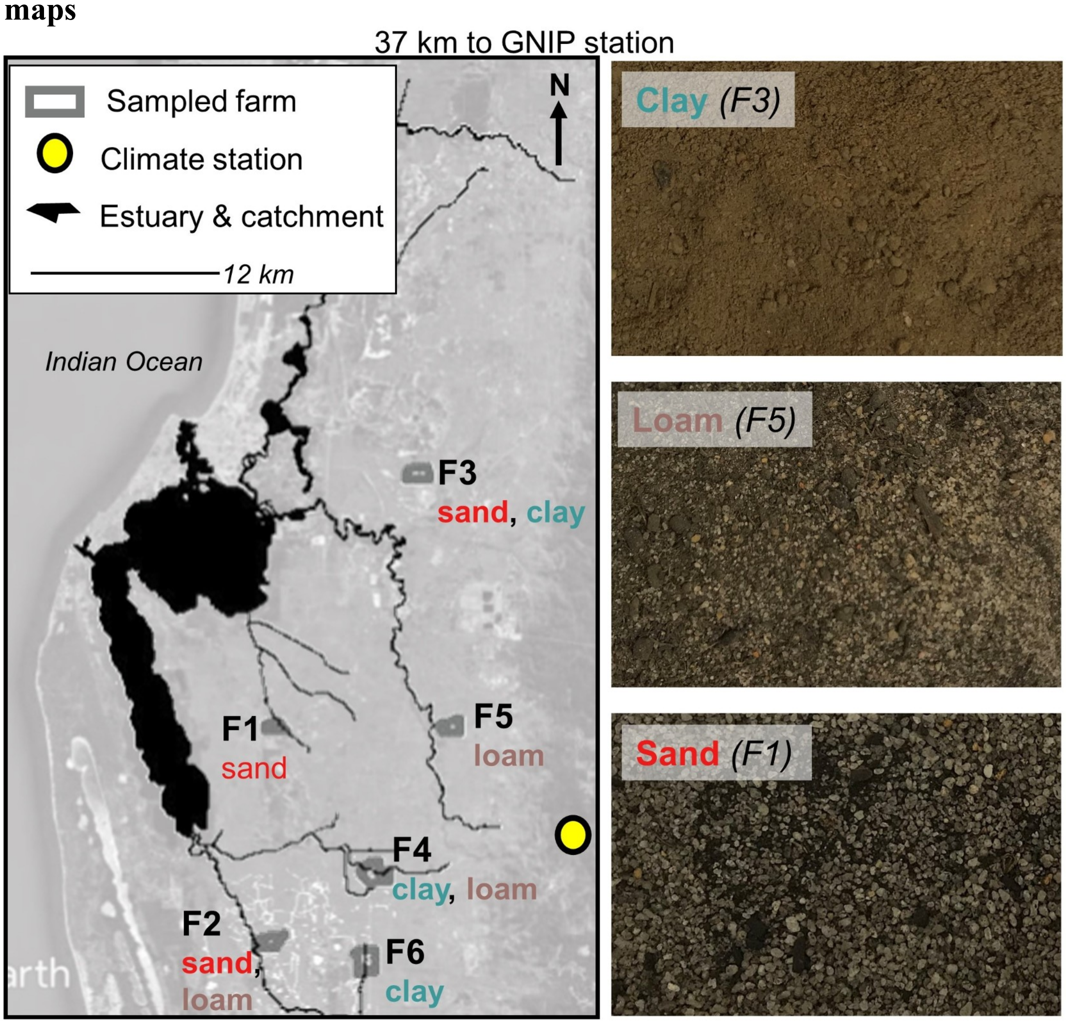

Fig. S1 Soil samples were collected from farms along the southern coast of Western Australia. Soils were collected from a total of 21 fields, which had either sand, loam or clay soil textures. Locations of the climate station used for long-term and daily soil temperature data is shown; the GNIP station used for long-term precipitation data is 37 km north of the mapped area. Base image on left panel from GoogleEarth. Soil photos from NS Wells.

The coverage of the three dominant soil textures in the Pinjarra (north-eastern) and Harvey (southern) portions of the catchment were used to test the sensitivity of $\delta^{18}O_{P(export)}$ signatures to realistic variations in soils over a catchment scale.

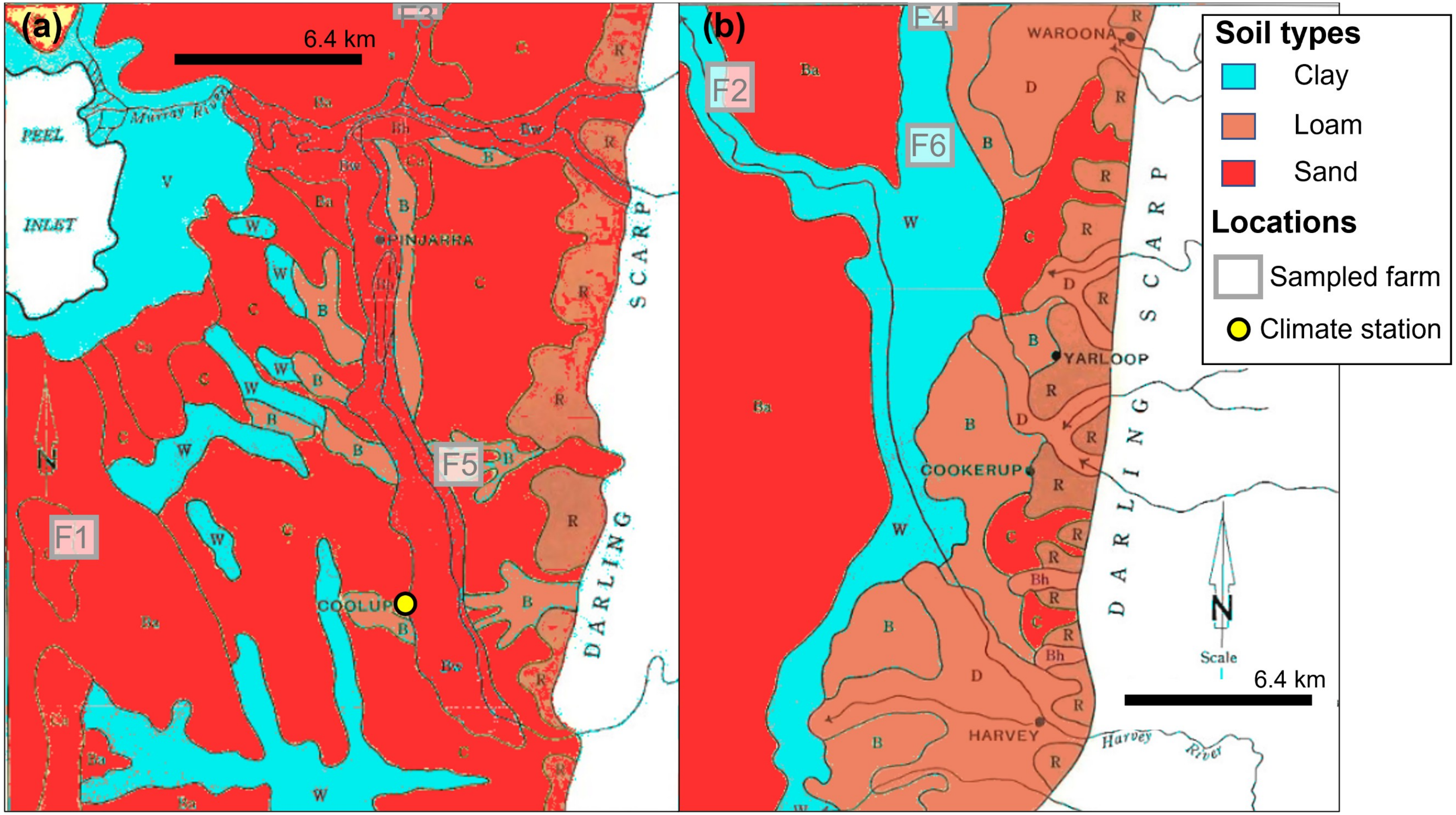

Fig. S2 Soil maps of the Pinjarra (a) and Harvey (b) sub-catchments, with approximate locations of farms sampled in this farm indicated. Zones are coloured based on soil texture (clay, loam, sand). Maps are from McArthur and Bettenay (1974), recoloured in R (Weller 2019). See original reference for detailed soil classifications.

S2: Additional soil data

Phosphorus mineralisation

The mineralisation of soil organic P forms to inorganic PO_4^{3-} is difficult to measure directly (Bünemann et al., 2015). However, it is important to parametrise in soil $\delta^{18}O_P$ studies as net mineralisation is a fractionating process. Here we estimated mineralisation rates using a modified version of the empirical equation developed by Achat et al. (2010):

(Eq. S1) $$P_{min\,(14)}=0.6976\left(P_{H2O}\cdot\frac{P_{org}}{P_{TIP}}\right)\cdot\left(1-e^{-0.0078}\right)+0.00026\cdot P_{org}\cdot 14$$

where $P_{min(14)}$ ($\mu g\ g^{-1}$) is the net amount of PO_4^{3-} mineralised over 14 days, estimated from the measured P_{org}. The original equation bases rates on microbial biomass P, which here was approximated by applying the ratio of P_{org} to P_{TIP} to the measured H_2O extractable PO_4^{3-} (P_{H2O}). Values generated from this equation should thus be treated as approximations rather than certainties.

Table S1: Additional soil P data. The phosphorus buffering index (PBI) and phosphorus environmental risk index (PERI) values are based on field sampling carried out prior to the presented work. Higher PBI values denote greater capacity to store P; higher PERI values denote greater likelihood of P export. Data from Rob Summers (Department of Primary Industries and Regional Development, Western Australia). See Burkitt et al. (2002) for details on index calculations. The P_{org}, TN, and N:P ratios were measured on samples collected during the presented July 2017 campaign, and are shown here to provide additional context on soil fertility differences.

Site	Soil texture	PBI	PERI	P_{org} *µg g⁻¹*	TN *µg g-¹*	N:P *g/g*
F1	Sand	1 – 4	2 – 10	220 (300)	320 (400)	0.95 (0.3)
F2	Loam	19 – 36	0.7	63 (20)	190 (31)	1.3 (0.3)
	Sand	3 – 5	2	72 (60)	290 (200)	2.7 (1)
F3	Clay	190	0.3	120 (40)	270 (70)	0.68 (0.1)
	Sand	1 – 2	2 – 4	41 (20)	170 (100)	2.6 (1)
F4	Clay	60 – 170	0.4	170 (70)	350 (100)	0.80 (0.1)
	Loam	50	0.7	120 (60)	310 (70)	1.0 (0.2)
F5	Loam	25 – 30	0.3 – 0.5	87 (30)	220 (70)	1.3 (0.2)
F6	Clay	90 – 100	0.1 – 0.2	120 (40)	330 (90)	1.3 (0.2)

Table S2: Sequential PO_4^{3-} extraction information for soils collected from pastures from six different farms (F1 – F6) across the Peel-Harvey catchment in Western Australia. 'Total' is the sum of the four sequential extractions, %leachable is the concentration of either H_2O extractable or H_2O+$NaHCO_3$ extractable PO_4^{3-} relative to the total, Soils are separated by dominant text (sand, loam, clay). Each sample (*n*) represents triplicate bulked soil cores (0-10 cm), which were each extracted in duplicate, and each extract analysed in duplicate.

Site	Soil texture	*n*	P-H_2O	P-$NaHCO_3$	P-NaOH	P-HCl	Total	% leachable (H_2O)	% leachable (H_2O+$NaHCO_3$)
F1	Sand	9	15 (10)	27 (30)	18 (20)	46 (50)	110 (100)	18 (10)	42 (20)
F2	Loam	6	7.4 (4)	16 (7)	16 (10)	6.3 (4)	45 (20)	20 (10)	55 (10)
	Sand	6	12 (9)	6.5 (4)	5.1 (4)	6.3 (6)	30 (20)	40 (20)	62 (20)
F3	Clay	3	0.24 (0.1)	34 (20)	130 (40)	48 (10)	210 (70)	0.11 (0.04)	16 (9)
	Sand	9	5.5 (4)	4.7 (3)	3.7 (1)	2.9 (3)	17 (5)	32 (20)	60 (20)
F4	Clay	9	3.4 (3)	25 (10)	120 (60)	59 (20)	210 (80)	1.5 (1)	13 (3)
	Loam	3	5.3 (4)	24 (10)	88 (50)	18 (10)	130 (80)	4.7 (5)	24 (10)
F5	Loam	9	2.3 (2)	9.7 (4)	18 (7)	1.9 (1)	32 (10)	7.4 (5)	37 (10)
F6	Clay	9	0.56 (0.6)	7.6 (4)	45 (9)	16 (9)	9.4 (70)	0.75 (0.7)	11 (5)

Table S3 $P_{min(14)}$ rates (Eq. S1) estimated for different pasture soil textures (clay, loam, sand) collected from six different farms within the coastal Peel-Harvey catchment (Western Australia). Values are mean (±SD)

Soil Texture	Farm	$P_{min(14)}$ $\mu g\ g^{-1}$
Clay	F3	0.049 (0.02)
	F4	0.15 (0.1)
	F6	0.061 (0.02)
Loam	F2	0.25 (0.2)
	F4	0.18 (0.1)
	F5	0.12 (0.07)
Sand	F1	0.40 (0.4)
	F2	0.55 (0.4)
	F3	0.27 (0.2)

S3: Input data for isotope mixing models

Table S4: The isotopic composition of rain collected across the farm sites within 1 month of collecting soils for $\delta^{18}O_{P(soil)}$ analysis.

Date	Amount (mm)	$\delta^{18}O_{H2O}$ (‰ v. VSMOW)
30/06/2017	3	-2.8
01/07/2017	2.6	0.6
01/07/2017	18	-5.8
03/07/2017	12	-2.7
11/07/2017	4.4	-6.3
11/07/2017	4.4	-2.3
21/07/2017	1.6	-2.5
26/07/2017	3	-0.9
28/07/2017	4.6	-2.7
08/08/2017		-5.3

Table S5 Temperature data used to calculate $\delta^{18}O_{P(eq)}$ values. $T_{soil(avg)}$ is from 86 years Medina Research Station data (http://www.bom.gov.au/climate/averages/tables/cw_009194.shtml); T_{air} is daily highs and lows for the sampled months (downloaded from the Bureau of Meteorology http://www.bom.gov.au/climate/dwo/IDCJDW6038.latest.shtml), and the expected offset between T_{soil} and T_{air} is based on data published in van den Hoogen et al. (2021).

Month	$T_{soil(avg)}$ °C	$T_{air(2017)}$ °C	Modelled offset (T_{soil} - T_{air})
January	22	-	-2.34
February	22.2	-	-0.0114
March	20.0	-	-0.553
April	16.4	-	-0.256
May	13.1	-	-0.213
June	11.2	8.8 – 23	-0.372
July	10.3	7.2 – 21	-0.704
August	10.6	5.7 – 26	-1.90
September	11.9	-	-3.44
October	14.1	-	-5.20
November	17.2	-	-2.57
December	20	-	-3.62
Annual average	**15.8**	**-**	**-1.8 ± 3**

Table S6: To get a better picture of likely daily variations in soil temperatures, we also used the Global Soil Microclimate Forecaster (Kearney 2019) to estimate temperatures at the soil surface (0 cm) and at 5 and 10 cm depth for winter the study area (Pinjarra, WA). Data can be accessed at: http://bioforecasts.science.unimelb.edu.au/app_direct/soil/

Time	$\mathbf{T_{0cm}}$	$\mathbf{T_{5cm}}$	$\mathbf{T_{10cm}}$
00:00	8.6	11	12
04:00	7.9	9.3	11
08:00	7.3	9.1	10
12:00	20	18	12
16:00	23	20	14
20:00	11	13	13
Daily Mean	**13 ± 7**	**13 ± 4**	**13 ± 2**

Table S7: The proportion of the total HCl extractable PO_4^{3-} pool of each soil texture that was likely to be recycled (X_P) was estimated over, 1) 1 hr, and, 2) >1 hr – 3 months, based on a 1:1 log-log relationships between exchangeable PO_4^{3-} and H_2O extractable PO_4^{3-} and NaOH extractable PO_4^{3-} (Helfenstein et al. 2020). Values are mean (±SD).

Soil texture	*n*	X_P **(1 hr)**	X_P **(~ 1 month)**
Clay	21	1.0 (1)	110 (40)
Loam	18	12 (10)	66 (40)
Sand	24	33 (20)	21 (9)

S4: R scripts for mixing models and up-scaling calculations

S4.1 Isotope equilibrium calculations

```
#formula for calculating d18Op(eq), just need to define a dataframe w temperature data and another
w d18OH2O data
#this handles converting precip data into long-term d18OH2O average for the location
d18Op_equilibrium<-function(df_temp,df_isotope) {


  #define factors
  e_evap<-3 #from Mathieu 1996, this should range between 3.5 and 5, ~3 for Sprenger et al 2017
  e_evap0<-0 #from Baenettin et al 2018, say mixing masks any evaporation signature

  #define temperature dataset & calculate long-term averages in C
  T_min<-mean(df_temp$Min_avg_soil_temp)
  T_max<-mean(df_temp$Max_avg_soil_temp)
  T_mean<-mean(df_temp$Avg_soil_temp)

  #calculate relevent d18OH2O range, including evaporative enrichment factor to correct precip to
soil
  (d18O_H2O_soil_mean<-sum(df_isotope$d18O_H2O_avg*df_isotope$Precipitation_mm_avg)/
    sum(df_isotope$Precipitation_mm_avg)+e_evap)
  (d18O_H2O_soil_mean0<-
sum(df_isotope$d18O_H2O_avg*df_isotope$Precipitation_mm_avg)/
    sum(df_isotope$Precipitation_mm_avg)+e_evap0)
  (d18O_H2O_soil_min<-sum((df_isotope$d18O_H2O_avg-
df_isotope$d18O_H2O_sd)*df_isotope$Precipitation_mm_avg)/
    sum(df_isotope$Precipitation_mm_avg)+e_evap)
  (d18O_H2O_soil_min0<-sum((df_isotope$d18O_H2O_avg-
df_isotope$d18O_H2O_sd)*df_isotope$Precipitation_mm_avg)/
    sum(df_isotope$Precipitation_mm_avg)+e_evap0)
  (d18O_H2O_soil_max<-
sum((df_isotope$d18O_H2O_avg+df_isotope$d18O_H2O_sd)*df_isotope$Precipitation_mm_avg)
/
    sum(df_isotope$Precipitation_mm_avg)+e_evap)
  (d18O_H2O_soil_max0<-
sum((df_isotope$d18O_H2O_avg+df_isotope$d18O_H2O_sd)*df_isotope$Precipitation_mm_avg)
/
    sum(df_isotope$Precipitation_mm_avg)+e_evap0)


  #create a table that summarises mean +-/- SD for soil T and water d18O for the site
  #create a table that summarises mean +-/- SD for soil T and water d18O for the site
  d18O_soil<-
rbind("d18O_H2O"=c(d18O_H2O_soil_mean,d18O_H2O_soil_min,d18O_H2O_soil_max,d18O_
H2O_soil_mean0,
                  d18O_H2O_soil_min0,d18O_H2O_soil_max0))

  T_soil<-rbind("T_soil"=c(T_mean,T_min,T_max))
```

```
    Parameter_summary<-expand.grid("d18O_H2O"=d18O_soil,"T_soil"=T_soil)

    #print(as.data.frame(Parameter_summary))

  #solve equilibration offset calculations using max, min, and mean temperature & d18O-H2O
values, & two parameterisations
  #equation 1 (Longinelli & Nuti 1973)


d18Oeq1_output=list((111.4-Parameter_summary$T_soil)/4.3+Parameter_summary$d18O_H2O)


  #equation 2 (Chang & Blake 2015 GCA) 1000ln*alpha=14.43*1000/T-26.54, where T is in K, so
d18Op = output+d18OH2O
  #rearrange to (d18OH2O + 1000)*e^((14.43*1000/T-26.54)/1000)-1000, see Hacker et al 2019
GCA

    d18Oeq2_output=list(

(Parameter_summary$d18O_H2O+1000)*exp((14.43*1000/(273.1+Parameter_summary$T_soil)-
26.3)/1000)-1000
    )


  d18Op_eq<-list("d18O_eq1"=unlist(d18Oeq1_output),"d18O_eq2"=unlist(d18Oeq2_output))

  #list2env(d18Op_eq,.GlobalEnv)



  #output data

  return(as.data.frame(d18Op_eq)
  )
}
```

S4.2 Mixing models

```
#load packages
library("grid")
library("plyr")
library("tidyverse")
library("munsell")
library("patchwork")

#import fertiliser data
Fertiliser_df<-read.csv("C:/All/Data/WA/PO4-isotopes/PO4iso-R/FertiliserDataPO4iso.csv")
View(Fertiliser_df)
#import soil data
Soils_df<-read.csv("C:/All/Data/WA/PO4-isotopes/PO4iso-R/WAsoils10.csv")
View(Soils_df)


#calculations box model based on different fertiliser mixing scenarios
#Model 1: immediate leaching (PO4=H2O extractable)
#Model 2: slower leaching (PO4=NaHCO3 extractable)
#Model 3: immediate leaching (PO4=H2O extractable) + fast equilibration (based on H2O pool)
#Model 4: slower leaching (PO4=NaHCO3 extractable) + slow equilibration (based on NaOH pool)

#input 1: fertiliser type (AG v SP d18O values)
#input 2: fertiliser application  rate (high v low range based on PBI of soil texture)
#fertilisation rate are in kg ha-1, so need to convert to ug g-1 of top 10 cm based on soil bulk
density.

#input 3: soil texture (P content, bulk density, PBI)
#input 4: equilibrium d18O value (use for soil+fert) & P turnover rate (Xp)

#output: d18Oleach values for each soil texture for each fert type 2013-2017



#define soil texture specific fertiliser rates
#depend on BD, recommended rates based on PBI range
#SoilType PBI_mean PBI_sd
#<chr>      <dbl>  <dbl>
#  1 Clay       116.    42.5
#for 90% yield: 90 58 48 37 25 14 2, so high = 58, low = 14, mean = 37
#2 Loam        30.4    9.95
#for 90% yield: 44 37 32 28 23 18 11, so high = 37, low = 18, mean = 28
#3 Sand         2.41   1.52
#for 90% yield: 16 13 9 4, so high = 13, low = 9, mean = 11
#rates from Summer & Weaver 2011 report on P app rates to clover pastures in WA


#implementing the model
#first, transform data into a series of nested lists based on soil texture
```

```
Soils_nested<- Soils_df%>%
  group_by(SoilType)%>%
  nest()


#second, define fertiliser functions specific for each soil texture


F_low_fun<-function(x) {

  BD_kgha = case_when(
    x=="Clay" ~ 1.44,
    x=="Loam"~1.24,
    x=="Sand"~1.33
  )

  F_low=case_when(
    x=="Clay" ~ 14*(10*10000*10000/1000)/(1000*1000*BD_kgha),
    x=="Loam" ~ 18*(10*10000*10000/1000)/(1000*1000*BD_kgha),
    x=="Sand" ~ 9*(10*10000*10000/1000)/(1000*1000*BD_kgha)
  )

  return(F_low)
}

F_high_fun<-function(x) {

  BD_kgha = case_when(
    x=="Clay" ~ 1.44,
    x=="Loam"~1.24,
    x=="Sand"~1.33
  )


  F_high=case_when(
    x=="Clay" ~ 58*(10*10000*10000/1000)/(1000*1000*BD_kgha),
    x=="Loam" ~ 37*(10*10000*10000/1000)/(1000*1000*BD_kgha),
    x=="Sand" ~ 13*(10*10000*10000/1000)/(1000*1000*BD_kgha)
  )

  return(F_high)
}

F_mean_fun<-function(x) {

  BD_kgha = case_when(
    x=="Clay" ~ 1.44,
    x=="Loam"~1.24,
    x=="Sand"~1.33
  )
```

```
  F_mean=case_when(
    x=="Clay" ~ 37*(10*10000*10000/1000)/(1000*1000*BD_kgha),
    x=="Loam" ~ 28*(10*10000*10000/1000)/(1000*1000*BD_kgha),
    x=="Sand" ~ 11*(10*10000*10000/1000)/(1000*1000*BD_kgha)
  )

  return(F_mean)
}

Soils_nested<-Soils_nested%>%
  mutate(F_low=map(SoilType,~{(F_low_fun(.))}))%>%
  mutate(F_high=map(SoilType,~{(F_high_fun(.))}))%>%
  mutate(F_mean=map(SoilType,~{(F_mean_fun(.))}))


#F application formula for scenario (a)
F_formula_a<-function(P_leach1,F_low,F_high,F_mean) {

  f_1=F_low/(P_leach1+F_low)
  f_2=F_high/(P_leach1+F_high)
  f_3=F_mean/(P_leach1+F_mean)

  x<-tibble::tibble(f_1,f_2,f_3) #this exports 3 columns of data
}

#F application formula for scenario (b)
F_formula_b<-function(P_leach2,F_low,F_high,F_mean) {

  f_1=F_low/(P_leach2+F_low)
  f_2=F_high/(P_leach2+F_high)
  f_3=F_mean/(P_leach2+F_mean)

  x<-tibble::tibble(f_1,f_2,f_3) #this works to export 4 columns of data, yay, where list etc didn't
}



#simplifies outputs info mean, min, max
F_formula2<-function(f_Fert) {

  mean_f=mean(f_Fert,na.rm=TRUE)
  low_f=mean(f_Fert,na.rm=TRUE)-sd(f_Fert,na.rm=TRUE)
  high_f=mean(f_Fert,na.rm=TRUE)+sd(f_Fert,na.rm=TRUE)
  x<-list(mean_f,low_f,high_f)

}

#solve d18O_soil range
```

```
d18O_formula<-function(d18O.PO4) {
  mean_d18O_s=mean(d18O.PO4,na.rm=TRUE)
  low_d18O_s=mean(d18O.PO4,na.rm=TRUE)-sd(d18O.PO4,na.rm=TRUE)
  high_d18O_s=mean(d18O.PO4,na.rm=TRUE)+sd(d18O.PO4,na.rm=TRUE)

  x<-list(mean_d18O_s,low_d18O_s,high_d18O_s)
  #how to make column instead of rows?
}


#use previously established formulas to solve (a) and (b)
Soils_nested<-Soils_nested%>%
  mutate(P_leach1=map(data,~.x$PO4_ug_g_H2O))%>%
  mutate(P_leach2=map(data,~.x$PO4_ug_g_H2O+.x$PO4_ug_g_NaHCO3))%>%
  mutate(f_range_a=pmap(list(P_leach1,F_low,F_high,F_mean),F_formula_a))%>%
  mutate(f_range_b=pmap(list(P_leach2,F_low,F_high,F_mean),F_formula_b))%>%
  mutate(f_Fert_a=map(f_range_a,rowMeans))%>%
  mutate(f_Fert_b=map(f_range_b,rowMeans))%>%
  mutate(f_summary_a=map(f_Fert_a,F_formula2))%>%
  mutate(f_summary_b=map(f_Fert_b,F_formula2))%>%
  mutate(d18O.PO4 = map(data,~.x$d18O.PO4))%>% #this extracts soil data
  mutate(d18O_soil=map(d18O.PO4,d18O_formula))#so now all soil data is ready



#now solve isotope leaching formula


#first create nested df so each mean d18O_fert value is a 'list' to be iterated over
Fertiliser_df<-Fertiliser_df%>%
  mutate(Year=parse_number(as.character(ID))
  )%>%
  mutate(Year=as.numeric(Year))%>%
  mutate(Year=Year+2000)
Fertiliser_df<-Fertiliser_df%>%
  mutate(Brand=case_when(
    grepl("AG",ID)~"AG",
    grepl("M",ID)~"MAP",
    grepl("S",ID)~"SP"
  ))

Fert_means<-Fertiliser_df%>%
  group_by(as.factor(Brand),Year)%>%
  summarise(
    d18O_mean=mean(d18O_PO4),
    d18O_sd=sd(d18O_PO4),
    d18O_min=min(d18O_PO4),
    d18O_max=max(d18O_PO4)
  )%>%
  rename(Brand="as.factor(Brand)")
```

```
Fert_means_nest<-Fert_means%>%
  dplyr::select(Brand,Year,d18O_mean)%>%
  filter(Brand!="MAP")%>% #remove MAP as only one year of data
  dplyr::group_by(Brand,Year)%>%
  nest()


#now separate data by soil texture

#for clays
{Soils_clay<-Soils_nested%>%
    dplyr::filter(SoilType=="Clay")%>%
    dplyr::mutate(f_summary_a=map(f_summary_a,~tibble(f_summary_a=.x)))%>%
    dplyr::mutate(f_summary_b=map(f_summary_b,~tibble(f_summary_b=.x)))%>%
    dplyr::mutate(d18O_soil=map(d18O_soil,~tibble(d18O_soil=.x)))%>%
    unnest(cols=c(f_summary_a,f_summary_b,d18O_soil))%>%
    dplyr::select(f_summary_a,f_summary_b,d18O_soil)%>%
    ungroup()
  Soils_clay$f_summary_a<-as.numeric(Soils_clay$f_summary_a)
  Soils_clay$f_summary_b<-as.numeric(Soils_clay$f_summary_b)
  Soils_clay$d18O_soil<-as.numeric(Soils_clay$d18O_soil)

  #generate all possible combinations of f v d18O_soil
  Soils_clay_a<-Soils_clay%>%tidyr::expand(f_summary_a,d18O_soil)
  Soils_clay_b<-Soils_clay%>%tidyr::expand(f_summary_b,d18O_soil)


  #now to expand so I can apply to the nested fertiliser df
  d18O_leach_formula<-function(d18O_mean) {

    #define equations for mixing
    formula_leach_a<-function(f_summary_a,d18O_soil) {
      f_summary_a*d18O_mean+(1-f_summary_a)*d18O_soil
    }

    formula_leach_b<-function(f_summary_b,d18O_soil) {
      f_summary_b*d18O_mean+(1-f_summary_b)*d18O_soil
    }

    #add equilibration formulas,based on Gross & Angert 2015 GCA:
    #%phosphate turnover = 100* (d18OP-MEAS - d18OP-RELES)/(d18OP-EQ - d18OP-RELES)
    #which modifies to: d18O_leach = %phos_turnover*(d18OP_eq-d18OP_Fert)+d18OP_Fert
    #w/in 1 day: 0 - 2 % turnover, so say 1% for H2O extractable (based on 1:1 log:log relationship
w H2O, see Helfenstein 2020)
    #w/in days-months: 70 - 150% turnover, so say 90% for NaHCO3 extractable

    formula_Leq_a<-function(d18O_leach_a) {

      Leq_1=0.06*(d18O_eq_low-d18O_leach_a)+d18O_leach_a
```

```
  Leq_2=0.06*(d18O_eq_high-d18O_leach_a)+d18O_leach_a
  Leq_3=0.06*(d18O_eq_mean-d18O_leach_a)+d18O_leach_a

  x=do.call(c,list(Leq_1,Leq_2,Leq_3)) #combines into one list

  return(x)
 }

 formula_Leq_b<-function(d18O_leach_b) {

  Leq_1=0.9*(d18O_eq_low-d18O_leach_b)+d18O_leach_b
  Leq_2=0.9*(d18O_eq_high-d18O_leach_b)+d18O_leach_b
  Leq_3=0.9*(d18O_eq_mean-d18O_leach_b)+d18O_leach_b

  x=do.call(c,list(Leq_1,Leq_2,Leq_3))

  return(x)
 }

 #define d18Op(eq) range, assuming generated mean+/-SD w d18O_eq2
 H2O_df<-winter_eq

 d18O_eq_low=H2O_df[2,]
 d18O_eq_high=H2O_df[3,]
 d18O_eq_mean=H2O_df[1,]

 #define mixing model terms
 f_summary_a=Soils_clay_a$f_summary_a
 f_summary_b=Soils_clay_b$f_summary_b
 d18O_soil=Soils_clay_a$d18O_soil


 #map equations to data
 d18O_leach_a=map2(f_summary_a,
             d18O_soil,
             formula_leach_a)

 d18O_leach_b=map2(f_summary_b,
             d18O_soil,
             formula_leach_b)

 d18O_Leq_a=map(d18O_leach_a,
           formula_Leq_a)
 d18O_Leq_b=map(d18O_leach_b,
           formula_Leq_b)


#then make outputs

 d18O_leach_min_a=min(unlist(d18O_leach_a))
```

```
    d18O_leach_max_a=max(unlist(d18O_leach_a))
    d18O_leach_mean_a=mean(unlist(d18O_leach_a))
    d18O_leach_sd_a=sd(unlist(d18O_leach_a))

    d18O_leach_min_b=min(unlist(d18O_leach_b))
    d18O_leach_max_b=max(unlist(d18O_leach_b))
    d18O_leach_mean_b=mean(unlist(d18O_leach_b))
    d18O_leach_sd_b=sd(unlist(d18O_leach_b))

    d18O_leach_min_c=min(unlist(d18O_Leq_a))
    d18O_leach_max_c=max(unlist(d18O_Leq_a))
    d18O_leach_mean_c=mean(unlist(d18O_Leq_a))
    d18O_leach_sd_c=sd(unlist(d18O_Leq_a))

    d18O_leach_min_d=min(unlist(d18O_Leq_b))
    d18O_leach_max_d=max(unlist(d18O_Leq_b))
    d18O_leach_mean_d=mean(unlist(d18O_Leq_b))
    d18O_leach_sd_d=sd(unlist(d18O_Leq_b))


    x<-tibble(d18O_leach_mean_a,d18O_leach_sd_a,d18O_leach_min_a,d18O_leach_max_a,
              d18O_leach_mean_b,d18O_leach_sd_b,d18O_leach_min_b,d18O_leach_max_b,
              d18O_leach_mean_c,d18O_leach_sd_c,d18O_leach_min_c,d18O_leach_max_c,
              d18O_leach_mean_d,d18O_leach_sd_d,d18O_leach_min_d,d18O_leach_max_d
    )

  }


  clay_leach<-Fert_means_nest%>%
    mutate(d18O_mean=map(data,~.x$d18O_mean))%>%
    mutate(map(d18O_mean,
               d18O_leach_formula))%>%
    unnest()
  }
#this works! now to make it a bit more universal so this is all less annoying

#for loams
{Soils_loam<-Soils_nested%>%
    dplyr::filter(SoilType=="Loam")%>%
    dplyr::mutate(f_summary_a=map(f_summary_a,~tibble(f_summary_a=.x)))%>%
    dplyr::mutate(f_summary_b=map(f_summary_b,~tibble(f_summary_b=.x)))%>%
    dplyr::mutate(d18O_soil=map(d18O_soil,~tibble(d18O_soil=.x)))%>%
    unnest(cols=c(f_summary_a,f_summary_b,d18O_soil))%>%
    dplyr::select(f_summary_a,f_summary_b,d18O_soil)%>%
    ungroup()
  Soils_loam$f_summary_a<-as.numeric(Soils_loam$f_summary_a)
  Soils_loam$f_summary_b<-as.numeric(Soils_loam$f_summary_b)
  Soils_loam$d18O_soil<-as.numeric(Soils_loam$d18O_soil)
```

```
  #generate all possible combinations of f v d18O_soil
  Soils_loam_a<-Soils_loam%>%tidyr::expand(f_summary_a,d18O_soil)
  Soils_loam_b<-Soils_loam%>%tidyr::expand(f_summary_b,d18O_soil)


  #now to expand so I can apply to the nested fertiliser df
  d18O_leach_formula_loam<-function(d18O_mean) {

    #define equations for mixing
    formula_leach_a<-function(f_summary_a,d18O_soil) {
      f_summary_a*d18O_mean+(1-f_summary_a)*d18O_soil
    }

    formula_leach_b<-function(f_summary_b,d18O_soil) {
      f_summary_b*d18O_mean+(1-f_summary_b)*d18O_soil
    }

    #add equilibration formulas,based on Gross & Angert 2015 GCA:
    #%phosphate turnover = 100* (d18OP-MEAS - d18OP-RELES)/(d18OP-EQ - d18OP-RELES)
    #which modifies to: d18O_leach = %phos_turnover*(d18OP_eq-d18OP_Fert)+d18OP_Fert
    #w/in 1 day: 2 - 22 % turnover, so say 10% for H2O extractable (based on 1:1 log:log
relationship w H2O, see Helfenstein 2020)
    #w/in days-months: 26 - 100% turnover, so say 60% for NaHCO3 extractable (based on NaOH,
as per above)

    formula_Leq_a<-function(d18O_leach_a) {

      Leq_1=0.1*(d18O_eq_low-d18O_leach_a)+d18O_leach_a
      Leq_2=0.1*(d18O_eq_high-d18O_leach_a)+d18O_leach_a
      Leq_3=0.1*(d18O_eq_mean-d18O_leach_a)+d18O_leach_a

      x=do.call(c,list(Leq_1,Leq_2,Leq_3)) #combines into one list

      return(x)
    }

    formula_Leq_b<-function(d18O_leach_b) {

      Leq_1=0.6*(d18O_eq_low-d18O_leach_b)+d18O_leach_b
      Leq_2=0.6*(d18O_eq_high-d18O_leach_b)+d18O_leach_b
      Leq_3=0.6*(d18O_eq_mean-d18O_leach_b)+d18O_leach_b

      x=do.call(c,list(Leq_1,Leq_2,Leq_3))

      return(x)
    }

    #define d18Op(eq) range, assuming generated mean+/-SD w d18O_eq2
    H2O_df<-winter_eq
```

```
d18O_eq_low=H2O_df[1,]
d18O_eq_high=H2O_df[3,]
d18O_eq_mean=H2O_df[2,]

#define mixing model terms
f_summary_a=Soils_loam_a$f_summary_a
f_summary_b=Soils_loam_b$f_summary_b
d18O_soil=Soils_loam_a$d18O_soil


#map equations to data
d18O_leach_a=map2(f_summary_a,
            d18O_soil,
            formula_leach_a)
d18O_leach_b=map2(f_summary_b,
            d18O_soil,
            formula_leach_b)
d18O_Leq_a=map(d18O_leach_a,
          formula_Leq_a)
d18O_Leq_b=map(d18O_leach_b,
          formula_Leq_b)


#then make outputs

d18O_leach_min_a=min(unlist(d18O_leach_a))
d18O_leach_max_a=max(unlist(d18O_leach_a))
d18O_leach_mean_a=mean(unlist(d18O_leach_a))
d18O_leach_sd_a=sd(unlist(d18O_leach_a))

d18O_leach_min_b=min(unlist(d18O_leach_b))
d18O_leach_max_b=max(unlist(d18O_leach_b))
d18O_leach_mean_b=mean(unlist(d18O_leach_b))
d18O_leach_sd_b=sd(unlist(d18O_leach_b))

d18O_leach_min_c=min(unlist(d18O_Leq_a))
d18O_leach_max_c=max(unlist(d18O_Leq_a))
d18O_leach_mean_c=mean(unlist(d18O_Leq_a))
d18O_leach_sd_c=sd(unlist(d18O_Leq_a))

d18O_leach_min_d=min(unlist(d18O_Leq_b))
d18O_leach_max_d=max(unlist(d18O_Leq_b))
d18O_leach_mean_d=mean(unlist(d18O_Leq_b))
d18O_leach_sd_d=sd(unlist(d18O_Leq_b))


x<-tibble(d18O_leach_mean_a,d18O_leach_sd_a,d18O_leach_min_a,d18O_leach_max_a,
       d18O_leach_mean_b,d18O_leach_sd_b,d18O_leach_min_b,d18O_leach_max_b,
       d18O_leach_mean_c,d18O_leach_sd_c,d18O_leach_min_c,d18O_leach_max_c,
       d18O_leach_mean_d,d18O_leach_sd_d,d18O_leach_min_d,d18O_leach_max_d
```

```
    )

  }


  loam_leach<-Fert_means_nest%>%
    mutate(d18O_mean=map(data,~.x$d18O_mean))%>%
    mutate(map(d18O_mean,
            d18O_leach_formula_loam))%>%
    unnest()
}

#for sand
{Soils_sand<-Soils_nested%>%
    dplyr::filter(SoilType=="Sand")%>%
    dplyr::mutate(f_summary_a=map(f_summary_a,~tibble(f_summary_a=.x)))%>%
    dplyr::mutate(f_summary_b=map(f_summary_b,~tibble(f_summary_b=.x)))%>%
    dplyr::mutate(d18O_soil=map(d18O_soil,~tibble(d18O_soil=.x)))%>%
    unnest(cols=c(f_summary_a,f_summary_b,d18O_soil))%>%
    dplyr::select(f_summary_a,f_summary_b,d18O_soil)%>%
    ungroup()
  Soils_sand$f_summary_a<-as.numeric(Soils_sand$f_summary_a)
  Soils_sand$f_summary_b<-as.numeric(Soils_sand$f_summary_b)
  Soils_sand$d18O_soil<-as.numeric(Soils_sand$d18O_soil)

  #generate all possible combinations of f v d18O_soil
  Soils_sand_a<-Soils_sand%>%tidyr::expand(f_summary_a,d18O_soil)
  Soils_sand_b<-Soils_sand%>%tidyr::expand(f_summary_b,d18O_soil)


  #now to expand so I can apply to the nested fertiliser df
  d18O_leach_formula_sand<-function(d18O_mean) {

    #define equations for mixing
    formula_leach_a<-function(f_summary_a,d18O_soil) {
      f_summary_a*d18O_mean+(1-f_summary_a)*d18O_soil
    }

    formula_leach_b<-function(f_summary_b,d18O_soil) {
      f_summary_b*d18O_mean+(1-f_summary_b)*d18O_soil
    }

    #add equilibration formulas,based on Gross & Angert 2015 GCA:
    #%phosphate turnover = 100* (d18OP-MEAS - d18OP-RELES)/(d18OP-EQ - d18OP-RELES)
    #which modifies to: d18O_leach = %phos_turnover*(d18OP_eq-d18OP_Fert)+d18OP_Fert
    #w/in 1 day: 13 - 53 % turnover, so say 30% for H2O extractable (based on 1:1 log:log
relationship w H2O, see Helfenstein 2020)
    #w/in days-months: 11 - 30% turnover, so say 20% for NaHCO3 extractable (based on NaOH
extractable w 1:1 log:log relationship, Helfenstein et al 2020)
```

```
formula_Leq_a<-function(d18O_leach_a) {

  Leq_1=0.3*(d18O_eq_low-d18O_leach_a)+d18O_leach_a
  Leq_2=0.3*(d18O_eq_high-d18O_leach_a)+d18O_leach_a
  Leq_3=0.3*(d18O_eq_mean-d18O_leach_a)+d18O_leach_a

  x=do.call(c,list(Leq_1,Leq_2,Leq_3)) #combines into one list

  return(x)
}

formula_Leq_b<-function(d18O_leach_b) {

  Leq_1=0.2*(d18O_eq_low-d18O_leach_b)+d18O_leach_b
  Leq_2=0.2*(d18O_eq_high-d18O_leach_b)+d18O_leach_b
  Leq_3=0.2*(d18O_eq_mean-d18O_leach_b)+d18O_leach_b

  x=do.call(c,list(Leq_1,Leq_2,Leq_3))

  return(x)
}

#define d18Op(eq) range, assuming generated mean+/-SD w d18O_eq2
H2O_df<-winter_eq

d18O_eq_low=H2O_df[1,]
d18O_eq_high=H2O_df[3,]
d18O_eq_mean=H2O_df[2,]

#define mixing model terms
f_summary_a=Soils_sand_a$f_summary_a
f_summary_b=Soils_sand_b$f_summary_b
d18O_soil=Soils_sand_a$d18O_soil


#map equations to data
d18O_leach_a=map2(f_summary_a,
            d18O_soil,
            formula_leach_a)
d18O_leach_b=map2(f_summary_b,
            d18O_soil,
            formula_leach_b)
d18O_Leq_a=map(d18O_leach_a,
          formula_Leq_a)
d18O_Leq_b=map(d18O_leach_b,
          formula_Leq_b)


#then make outputs
```

```
    d18O_leach_min_a=min(unlist(d18O_leach_a))
    d18O_leach_max_a=max(unlist(d18O_leach_a))
    d18O_leach_mean_a=mean(unlist(d18O_leach_a))
    d18O_leach_sd_a=sd(unlist(d18O_leach_a))

    d18O_leach_min_b=min(unlist(d18O_leach_b))
    d18O_leach_max_b=max(unlist(d18O_leach_b))
    d18O_leach_mean_b=mean(unlist(d18O_leach_b))
    d18O_leach_sd_b=sd(unlist(d18O_leach_b))

    d18O_leach_min_c=min(unlist(d18O_Leq_a))
    d18O_leach_max_c=max(unlist(d18O_Leq_a))
    d18O_leach_mean_c=mean(unlist(d18O_Leq_a))
    d18O_leach_sd_c=sd(unlist(d18O_Leq_a))

    d18O_leach_min_d=min(unlist(d18O_Leq_b))
    d18O_leach_max_d=max(unlist(d18O_Leq_b))
    d18O_leach_mean_d=mean(unlist(d18O_Leq_b))
    d18O_leach_sd_d=sd(unlist(d18O_Leq_b))


    x<-tibble(d18O_leach_mean_a,d18O_leach_sd_a,d18O_leach_min_a,d18O_leach_max_a,
          d18O_leach_mean_b,d18O_leach_sd_b,d18O_leach_min_b,d18O_leach_max_b,
          d18O_leach_mean_c,d18O_leach_sd_c,d18O_leach_min_c,d18O_leach_max_c,
          d18O_leach_mean_d,d18O_leach_sd_d,d18O_leach_min_d,d18O_leach_max_d
    )

  }


  sand_leach<-Fert_means_nest%>%
    mutate(d18O_mean=map(data,~.x$d18O_mean))%>%
    mutate(map(d18O_mean,
           d18O_leach_formula_sand))%>%
    unnest()
}


#collate and organise outputs
#try making boxplots instead as more accurate for changes over years, need to
#pivot longer to collate mean, min, max
{
  #first add columns for +- SD
  sand_leach<-sand_leach%>%
    mutate(mean_a_1=d18O_leach_mean_a-d18O_leach_sd_a,
        mean_a_2=d18O_leach_mean_a+d18O_leach_sd_a,
        mean_b_1=d18O_leach_mean_b-d18O_leach_sd_b,
        mean_b_2=d18O_leach_mean_b+d18O_leach_sd_b,
        mean_c_1=d18O_leach_mean_c-d18O_leach_sd_c,
        mean_c_2=d18O_leach_mean_c+d18O_leach_sd_c,
```

```
       mean_d_1=d18O_leach_mean_d-d18O_leach_sd_d,
       mean_d_2=d18O_leach_mean_d+d18O_leach_sd_d)
clay_leach<-clay_leach%>%
  mutate(mean_a_1=d18O_leach_mean_a-d18O_leach_sd_a,
       mean_a_2=d18O_leach_mean_a+d18O_leach_sd_a,
       mean_b_1=d18O_leach_mean_b-d18O_leach_sd_b,
       mean_b_2=d18O_leach_mean_b+d18O_leach_sd_b,
       mean_c_1=d18O_leach_mean_c-d18O_leach_sd_c,
       mean_c_2=d18O_leach_mean_c+d18O_leach_sd_c,
       mean_d_1=d18O_leach_mean_d-d18O_leach_sd_d,
       mean_d_2=d18O_leach_mean_d+d18O_leach_sd_d)
loam_leach<-loam_leach%>%
  mutate(mean_a_1=d18O_leach_mean_a-d18O_leach_sd_a,
       mean_a_2=d18O_leach_mean_a+d18O_leach_sd_a,
       mean_b_1=d18O_leach_mean_b-d18O_leach_sd_b,
       mean_b_2=d18O_leach_mean_b+d18O_leach_sd_b,
       mean_c_1=d18O_leach_mean_c-d18O_leach_sd_c,
       mean_c_2=d18O_leach_mean_c+d18O_leach_sd_c,
       mean_d_1=d18O_leach_mean_d-d18O_leach_sd_d,
       mean_d_2=d18O_leach_mean_d+d18O_leach_sd_d)

sand_leach_long<-sand_leach%>%
  dplyr::select(-contains("sd"))%>% #removes the sd columns
  dplyr::select(-c(d18O_mean,d18O_mean1))%>%
  pivot_longer(cols=!c("Brand","Year"),
             names_to="stat",values_to="d18Oleach")
sand_leach_long<-sand_leach_long%>%
  mutate(scenario=case_when(
    endsWith(stat,"a") ~ "a",
    endsWith(stat,"b")~"b",
    endsWith(stat,"c")~"c",
    endsWith(stat,"d")~"d"
  ))%>%
  mutate(Soil_Type="sand")
clay_leach_long<-clay_leach%>%
  dplyr::select(-contains("sd"))%>% #removes the sd columns
  dplyr::select(-c(d18O_mean,d18O_mean1))%>%
  pivot_longer(cols=!c("Brand","Year"),
             names_to="stat",values_to="d18Oleach")
clay_leach_long<-clay_leach_long%>%
  mutate(scenario=case_when(
    endsWith(stat,"a") ~ "a",
    endsWith(stat,"b")~"b",
    endsWith(stat,"c")~"c",
    endsWith(stat,"d")~"d"
  ))%>%
  mutate(Soil_Type="clay")
loam_leach_long<-loam_leach%>%
  dplyr::select(-contains("sd"))%>% #removes the sd columns
  dplyr::select(-c(d18O_mean,d18O_mean1))%>%
```

```
  pivot_longer(cols=!c("Brand","Year"),
               names_to="stat",values_to="d18Oleach")
 loam_leach_long<-loam_leach_long%>%
   mutate(scenario=case_when(
     endsWith(stat,"a") ~ "a",
     endsWith(stat,"b")~"b",
     endsWith(stat,"c")~"c",
     endsWith(stat,"d")~"d"
   ))%>%
   mutate(Soil_Type="loam")
}

#upscale to see what can be picked out v not at catchment scale
#use model estimates of %P from each soil texture, and then just vary over time

leach_all_long<-rbind(sand_leach_long,loam_leach_long,clay_leach_long)
write.csv(leach_all_long,"leach_all_long.csv")
leach_SP_long<-leach_all_long%>%
  dplyr::filter(Brand=="SP")
leach_AG_long<-leach_all_long%>%
  dplyr::filter(Brand=="AG")
#or nonpivoted
sand_leach<-sand_leach%>%
  mutate(Soil_Type="sand")
loam_leach<-loam_leach%>%
  mutate(Soil_Type="loam")
clay_leach<-clay_leach%>%
  mutate(Soil_type="clay")
leach_all<-rbind(sand_leach, loam_leach,clay_leach)
leach_SP<-leach_all%>%
  dplyr::filter(Brand=="SP")
leach_AG<-leach_all%>%
  dplyr::filter(Brand=="AG")
write.csv(leach_all,"leach_all_wide.csv")

#summary stats

Soils_clay_a%>%
  summarise(
    avg_f=mean(f_summary_a),
    sd_f=sd(f_summary_a)
  )

Soils_clay_b%>%
  summarise(
    avg_f=mean(f_summary_b),
    sd_f=sd(f_summary_b)
  )

Soils_loam_b%>%
```

```
  summarise(
    avg_f=mean(f_summary_b),
    sd_f=sd(f_summary_b)
  )

Soils_sand_b%>%
  summarise(
    avg_f=mean(f_summary_b),
    sd_f=sd(f_summary_b)
  )



#Plot data

#sort out aesthetics
{
  PERI_colours<-c(mnsl("7.5BG 5/6"),mnsl("7.5R 5/4"),mnsl("7.5R 5/18")) #clay #loam #sand
  clay_colour<-mnsl("7.5BG 5/6")
  clay_colour2<-mnsl("7.5BG 8/6")
  loam_colour<-mnsl("7.5R 5/4")
  loam_colour2<-mnsl("7.5R 4/4")
  sand_colour<-mnsl("7.5R 5/18")

  #colours for fert brands
  Fert_colours2<-c(mnsl("N 0/0"),mnsl("N 7/0"))
}


#plots w separated soil texture
{  #plot f_summary values for each soil texture for reference
  f_a<-ggplot()+
    geom_violin(data=
               Soils_nested%>% #extract relevant data
               unnest(f_range_a)%>%
               dplyr::select(SoilType,f_1,f_2,f_3)%>%
               pivot_longer(cols = c(f_1,f_2,f_3)),
             aes(x=SoilType,y=value,fill=SoilType),alpha=0.8,
             draw_quantiles = 0.5,scale = "area",
             adjust=0.8, colour="black")+
    scale_colour_manual(values=PERI_colours)+
    scale_fill_manual(values=PERI_colours)+
    annotate("text",x=3.4,y=0.9,label="(a)",size=4)+
    scale_y_continuous(limits=c(0,1),expand=c(0,0),
                  breaks=c(0,0.5,1.0))+
    labs(y=expression(italic("f")["fert"]))+
    theme_classic()+
    theme(legend.position="none",axis.title.x = element_blank(),
       text=element_text(size=16))+
    annotation_custom(grob=grid::textGrob(label="fast",
```

```
                              gp=grid::gpar(fontsize=18,fontface="italic")),
                    xmin=5,xmax=5,ymin=0.5,ymax=0.7)+
  coord_cartesian(clip="off")

#plot f_summary values for each soil texture for reference
f_b<-ggplot()+
  geom_violin(data=
              Soils_nested%>% #extract relevant data
              unnest(f_range_b)%>%
              dplyr::select(SoilType,f_1,f_2,f_3)%>%
              pivot_longer(cols = c(f_1,f_2,f_3)),
             aes(x=SoilType,y=value,fill=SoilType),alpha=0.8,
             draw_quantiles = 0.5,scale = "area",
             adjust=0.8, colour="black")+
  scale_colour_manual(values=PERI_colours)+
  scale_fill_manual(values=PERI_colours)+
  annotate("text",x=3.4,y=0.9,label="(b)",size=4)+
  scale_y_continuous(limits=c(0,1),expand=c(0,0),
                     breaks=c(0,0.5,1.0))+
  labs(y=expression(italic("f")["fert"]))+
  theme_classic()+
  theme(legend.position="none",axis.title.x = element_blank(),
        text=element_text(size=16))+
  annotation_custom(grob=grid::textGrob(label="slow",
                              gp=grid::gpar(fontsize=18,fontface="italic")),
                    xmin=5,xmax=5,ymin=0.5,ymax=0.7)+
  coord_cartesian(clip="off")

#for scenario (a), immediate leaching but with no turnover
iso_a_2<-ggplot()+
  geom_boxplot(data=leach_AG_long%>%filter(scenario=="a"),
             aes(x=as.factor(Year),y=d18Oleach,fill=Soil_Type),
             colour=mnsl("N 0/0"),alpha=.9,size=.6,linetype="solid")+
  geom_boxplot(data=leach_SP_long%>%filter(scenario=="a"),
             aes(x=as.factor(Year),y=d18Oleach,fill=Soil_Type),
             colour=mnsl("N 7/0"),linetype="longdash",alpha=0.7,
             size=.6)+
  geom_point(data=Fert_means%>%dplyr::filter(Brand!="MAP"),aes(
    x=as.factor(Year),y=d18O_mean,shape=Brand,colour=Brand),size=3)+
  geom_linerange(data=Fert_means%>%dplyr::filter(Brand!="MAP"),
               aes(x=as.factor(Year),ymin=d18O_mean-d18O_sd,ymax=d18O_mean+d18O_sd,
                   colour=Brand))+
  annotate("text",x="2013",y=24.5,label=expression(italic("X")["P"]*~"= nil"))+
  annotate("text",x="2017",y=25,label="(c)",size=4)+
  scale_y_continuous(limits=c(15,25.5),expand=c(0,0),breaks=c(15,18,21,24))+
  scale_colour_manual(values=Fert_colours2)+
  scale_fill_manual(values=PERI_colours)+
  labs(y=expression(delta^"18"*"O"["P(export-1)"]*~"(‰)"))+
  theme_classic()+
  theme(legend.title=element_blank(),text=element_text(size=18),
```

```
      legend.position = "none",axis.title.y = element_text(size=16),
      axis.text.x = element_blank(),axis.title.x=element_blank())


#for scenario (b), seasonal leaching but with no turnover
iso_b_2<-ggplot()+
  geom_boxplot(data=leach_AG_long%>%filter(scenario=="b"),
             aes(x=as.factor(Year),y=d18Oleach,fill=Soil_Type),
             colour=mnsl("N 0/0"),alpha=.9,size=.6,linetype="solid")+
  geom_boxplot(data=leach_SP_long%>%filter(scenario=="b"),
             aes(x=as.factor(Year),y=d18Oleach,fill=Soil_Type),
             colour=mnsl("N 7/0"),linetype="longdash",alpha=0.7,
             size=.6)+
  geom_point(data=Fert_means%>%dplyr::filter(Brand!="MAP"),aes(
    x=as.factor(Year),y=d18O_mean,shape=Brand,colour=Brand),size=3)+
  geom_linerange(data=Fert_means%>%dplyr::filter(Brand!="MAP"),
               aes(x=as.factor(Year),ymin=d18O_mean-d18O_sd,ymax=d18O_mean+d18O_sd,
                   colour=Brand))+
  annotate("text",x="2013",y=24.5,label=expression(italic("X")["P"]*~"= nil"))+
  annotate("text",x="2017",y=25,label="(d)",size=4)+
  scale_y_continuous(limits=c(15,25.5),expand=c(0,0),breaks=c(15,18,21,24))+
  scale_colour_manual(values=Fert_colours2)+
  scale_fill_manual(values=PERI_colours)+
  labs(y=expression(delta^"18"*"O"["P(export)"]*~"(‰ v. VSMOW)"))+
  theme_classic()+
  theme(legend.title=element_blank(),text=element_text(size=16),
      legend.position = "none",axis.title=element_blank(),
      axis.text=element_blank())


#for scenario (c), immediate leaching but with ~1 hr of turnover
iso_c_2<-ggplot()+
  geom_boxplot(data=leach_AG_long%>%filter(scenario=="c"),
             aes(x=as.factor(Year),y=d18Oleach,fill=Soil_Type),
             colour=mnsl("N 0/0"),alpha=.9,size=.6,linetype="solid")+
  geom_boxplot(data=leach_SP_long%>%filter(scenario=="c"),
             aes(x=as.factor(Year),y=d18Oleach,fill=Soil_Type),
             colour=mnsl("N 7/0"),linetype="longdash",alpha=0.7,
             size=.6)+
  geom_point(data=Fert_means%>%dplyr::filter(Brand!="MAP"),aes(
    x=as.factor(Year),y=d18O_mean,shape=Brand,colour=Brand),size=3)+
  geom_linerange(data=Fert_means%>%dplyr::filter(Brand!="MAP"),
               aes(x=as.factor(Year),ymin=d18O_mean-d18O_sd,ymax=d18O_mean+d18O_sd,
                   colour=Brand))+
  annotate("text",x="2013",y=24.5,label=expression(italic("X")["P"]*~"= 1 hr"))+
  annotate("text",x="2017",y=25,label="(e)",size=4)+
  scale_y_continuous(limits=c(15,25.5),expand=c(0,0),breaks=c(15,18,21,24))+
  scale_colour_manual(values=Fert_colours2)+
  scale_fill_manual(values=PERI_colours)+
  labs(y=expression(delta^"18"*"O"["P(export-2)"]*~"(‰)"),
```

```
      x=expression("Manufacturing year"))+
  theme_classic()+
  theme(legend.title=element_blank(),text=element_text(size=16),
        legend.position = "none",axis.title.y = element_text(size=16),
        axis.title.x=element_blank())

#for scenario (d), seasonal leaching but with hourly - seasonal turnover
iso_d_2<-ggplot()+
  geom_boxplot(data=leach_AG_long%>%filter(scenario=="d"),
               aes(x=as.factor(Year),y=d18Oleach,fill=Soil_Type),
               colour=mnsl("N 0/0"),alpha=.9,size=.6,linetype="solid")+
  geom_boxplot(data=leach_SP_long%>%filter(scenario=="d"),
               aes(x=as.factor(Year),y=d18Oleach,fill=Soil_Type),
               colour=mnsl("N 7/0"),linetype="longdash",alpha=0.7,
               size=.6)+
  geom_point(data=Fert_means%>%dplyr::filter(Brand!="MAP"),aes(
    x=as.factor(Year),y=d18O_mean,shape=Brand,colour=Brand),size=3)+
  geom_linerange(data=Fert_means%>%dplyr::filter(Brand!="MAP"),
                 aes(x=as.factor(Year),ymin=d18O_mean-d18O_sd,ymax=d18O_mean+d18O_sd,
                     colour=Brand))+
  annotate("text",x="2013",y=24.5,label=expression(italic("X")["P"]*~"= 1 month"))+
  annotate("text",x="2017",y=25,label="(f)",size=4)+
  scale_y_continuous(limits=c(15,25.5),expand=c(0,0),breaks=c(15,18,21,24))+
  scale_colour_manual(values=Fert_colours2)+
  scale_fill_manual(values=PERI_colours)+
  labs(y=expression(delta^"18"*"O"["P(export)"]*~"(‰ v. VSMOW)"))+
  theme_classic()+
  theme(legend.title=element_blank(),text=element_text(size=16),
        legend.position = "none",axis.title.y = element_blank(),
        axis.text.y = element_blank(),axis.title.x=element_blank())


#combine into one model figure
(f_a+plot_spacer()|f_b+plot_spacer())/
  (iso_a_2|iso_b_2)/
  (iso_c_2|iso_d_2)+
  plot_annotation(theme=theme(plot.margin=margin(0,0,0,0,unit="pt")))+
  wrap_elements(grid::textGrob(label="Fertiliser year",
                               gp=grid::gpar(fontsize=15)),
                clip=FALSE)+
  plot_layout(heights=(c(1,2,2,0.3)))

}
```

S4.3 Up-scaling calculations

```
#to calculate the 'catchment scale' variations that would be caused by
#different %fertiliser use, & proportion of fast v slow leaching (both w/ turnover as this is most
#likely)
#P load is 145 t y-1, catchment area is 11930 km2 (if including all the way up into the Wheatbelt),
#50% of P from beef grazing
#in coastal plain catchment is ~1,800 km2 and ~1,100 km2 is beef grazing
#but here we're ignoring land-use and just focusing on soil texture to calculate need %P from sandy
v #loam v clay

#install packages
library("countcolors")

#figure out soil coverage: key for maps from McArthur 1974 CSIRO maps of soils
#W = Wellesley = Clay
#R = Ridge Hill Shelf = Basically rock, exclude
#C = Coolup = Sand
#B = Boyanup = Loam
#Bh = Belhus = Sand
#D = Dardanup = Loam
#Ba = Bassendean = Sand
#Bw = Blythewood = Sand
#Ca = Cannington = Sand
#S = Spearwood = Sand
#V = Vasse = Clay

#bring in pinjarra (Murray River catchment) map
pinjarra<-jpeg::readJPEG("pinjarra.jpg")
#identify colour ranges
colordistance::plotPixels("pinjarra.jpg",lower=NULL,upper=NULL,n=5000)
clusters.pinjarra<-colordistance::getKMeanColors("pinjarra.jpg",n=13)
colordistance::extractClusters(clusters.pinjarra)

#then define values / percents
colour_1<-c(0.8834396,0.7051185,0.6842976)
pinjarra_1<-countcolors::sphericalRange(pinjarra,center=colour_1,radius=0.07,
                              target.color="red"); names(pinjarra_1)
Bw<-pinjarra_1$img.fraction
colour_2<-c(0.97,0.51,0.5)
pinjarra_2<-countcolors::sphericalRange(pinjarra,center=colour_2,radius=0.06,
                              target.color="red"); names(pinjarra_2)
Bh<-pinjarra_2$img.fraction
colour_3<-c(0.9919349,0.9900777,0.76423475)
pinjarra_3<-countcolors::sphericalRange(pinjarra,center=colour_3,radius=0.07,
                              target.color="red"); names(pinjarra_3)
Ba<-pinjarra_3$img.fraction
colour_4<-c(0.8808570,0.9222708,0.5781587)
pinjarra_4<-countcolors::sphericalRange(pinjarra,center=colour_4,radius=0.07,
```

```
                          target.color="red"); names(pinjarra_4)
C<-pinjarra_4$img.fraction
colour_5<-c(0.4587057,0.4162613,0.2991963) #line colour, ignore
pinjarra_5<-countcolors::sphericalRange(pinjarra,center=colour_5,radius=0.07,
                          target.color="red"); names(pinjarra_5)
colour_6<-c(0.9429240,0.7212089,0.5159612)
pinjarra_6<-countcolors::sphericalRange(pinjarra,center=colour_6,radius=0.07,
                          target.color="red"); names(pinjarra_6)
Ca<-pinjarra_6$img.fraction
colour_7<-c(0.2248027,0.2046666,0.1042162)
pinjarra_7<-countcolors::sphericalRange(pinjarra,center=colour_7,radius=0.07,
                          target.color="red"); names(pinjarra_7)
#line colour, ignore
colour_8<-c(0.6892578,0.6781974,0.65941717)
pinjarra_8<-countcolors::sphericalRange(pinjarra,center=colour_8,radius=0.08,
                          target.color="red"); names(pinjarra_8)
W<-pinjarra_8$img.fraction
colour_9<-c(0.9974429,0.9975196,0.99377882)
pinjarra_9<-countcolors::sphericalRange(pinjarra,center=colour_9,radius=0.07,
                          target.color="red"); names(pinjarra_9)
#white space, but struggling with V
colour_10<-c(0.8918985,0.9560049,0.92361884)
pinjarra_10<-countcolors::sphericalRange(pinjarra,center=colour_10,radius=0.05,
                          target.color="red"); names(pinjarra_10)
V<-pinjarra_10$img.fraction
colour_11<-c(0.7156843,0.8310881,0.71357728)
pinjarra_11<-countcolors::sphericalRange(pinjarra,center=colour_11,radius=0.05,
                          target.color="red"); names(pinjarra_11)
B_p<-pinjarra_11$img.fraction

#make recoloured figure for SI
loam_centre<-colour_11
clay_centre.1<-colour_10
clay_centre.2<-colour_8
sand_centre.1<-colour_6
sand_centre.5<-colour_4
sand_centre.2<-colour_3
sand_centre.3<-colour_2
sand_centre.4<-colour_1
bkg_centre<-c(0.7066147,0.8467714,0.70303157)

pinjarra_map_coloured<-countcolors::countColors("pinjarra.jpg",color.range = "spherical",
                                  center=c(loam_centre,clay_centre.1,
                                        clay_centre.2,sand_centre.4,
                                        sand_centre.1,sand_centre.3,
                                        sand_centre.2,sand_centre.5),
                                radius=c(0.07,0.05,0.08,0.07,0.07,0.07,0.07,0.07),
                                plotting=TRUE,bg.lower = bkg_centre,
                                  target.color=c("salmon2","cyan2",
                                            "cyan2","firebrick1",
```

```
                                          "firebrick1","firebrick1",
                                          "firebrick1","firebrick1"))



#normalise area to soil area
pinjarra_soils<-sum(B_p,V,W,Ca,C,Ba,Bh,Bw)
pinjarra_clay<-(W+V)/pinjarra_soils
pinjarra_loam<-B_p/pinjarra_soils
pinjarra_sand<-(Ca+C+Ba+Bh+Bw)/pinjarra_soils
pinjarra_sand

#bring in harvey (Harvey River catchment) map
harvey<-jpeg::readJPEG("harvey3.jpg")
#identify colour ranges
colordistance::plotPixels("harvey3.jpg",lower=NULL,upper=NULL,n=6000)
clusters.harvey<-colordistance::getKMeanColors("harvey3.jpg",n=10)
colordistance::extractClusters(clusters.harvey)

#then define values / percents
H_1<-c(0.9799629,0.5392124,0.4162087)
harvey_1<-countcolors::sphericalRange(harvey,center=H_1,radius=0.04,
                           target.color="red"); names(harvey_1)
D_H<-harvey_1$img.fraction
H_2<-c(0.6634102,0.6426091,0.6494207)
harvey_2<-countcolors::sphericalRange(harvey,center=H_2,radius=0.07,
                          target.color="red"); names(harvey_2)
W_H<-harvey_2$img.fraction
H_4<-c(0.9952840,0.9960592,0.7311766)
harvey_4<-countcolors::sphericalRange(harvey,center=H_4,radius=0.07,
                          target.color="red"); names(harvey_4)
Ba_H<-harvey_4$img.fraction
H_5<-c(0.8437691,0.5067290,0.3829869)
harvey_5<-countcolors::sphericalRange(harvey,center=H_5,radius=0.06,
                          target.color="red"); names(harvey_5)
R_H<-harvey_5$img.fraction
H_6<-c(0.8828925,0.8986511,0.5540137)
harvey_6<-countcolors::sphericalRange(harvey,center=H_6,radius=0.07,
                          target.color="red"); names(harvey_6)
C_H<-harvey_6$img.fraction
H_7<-c(0.6770744,0.8317865,0.6913081)
harvey_7<-countcolors::sphericalRange(harvey,center=H_7,radius=0.07,
                          target.color="red"); names(harvey_7)
B_H<-harvey_7$img.fraction
H_8<-c(0.4427807,0.4320241,0.3742824)
harvey_8<-countcolors::sphericalRange(harvey,center=H_8,radius=0.07,
                          target.color="red"); names(harvey_8)
#lines, ignore
H_9<-c(0.5658113,0.2799842,0.1750493)
harvey_9<-countcolors::sphericalRange(harvey,center=H_9,radius=0.07,
```

```
                              target.color="red"); names(harvey_9)
#lines, ignore
H_10<-c(0.2151338,0.1874336,0.1253385)
harvey_10<-countcolors::sphericalRange(harvey,center=H_10,radius=0.07,
                              target.color="red"); names(harvey_10)

#normalise area to soil area
harvey_soils<-sum(D_H,W_H,Ba_H,C_H,B_H)
harvey_clay<-(W_H)/harvey_soils
harvey_loam<-(B_H+D_H)/harvey_soils
harvey_sand<-(Ba_H+C_H)/harvey_soils

#make recoloured figure for SI
loam_H1<-H_7
loam_H2<-H_1
clay_H1<-H_2
sand_H1<-H_4
sand_H2<-H_6

harvey_map_coloured<-countcolors::countColors("harvey3.jpg",color.range = "spherical",
                                    center=c(loam_H1,loam_H2,
                                          clay_H1,sand_H1,
                                          sand_H2),
                                    radius=c(0.07,0.07,0.07,0.07,0.07),
                                    plotting=TRUE,bg.lower = bkg_centre,
                                    target.color=c("salmon2","salmon2",
                                             "cyan2","firebrick1",
                                             "firebrick1"))




#so then to apply this to P data, need:
#1) Pfert+Psoil for fast v slow for each soil texture
#2) d18Op for scenarios a v d for each fertiliser & soil texture

#then calculate 'leachable P content' (as P_leach+Fert) for each soil texture
P_a_formula<-function(P_leach1,F_low,F_high,F_mean) {

  P_1=(P_leach1+F_low)
  P_2=(P_leach1+F_high)
  P_3=(P_leach1+F_mean)

  P_all<-tibble::tibble(P_1,P_2,P_3) #table w three columns

  P_summary=rowMeans(P_all) #makes one column

  mean_P=mean(P_summary,na.rm=TRUE)
  low_P=mean(P_summary,na.rm=TRUE)-sd(P_summary,na.rm=TRUE)
  high_P=mean(P_summary,na.rm=TRUE)+sd(P_summary,na.rm=TRUE)
```

```
  x<-c(mean_P,low_P,high_P)

  return(x)

}

P_b_formula<-function(P_leach2,F_low,F_high,F_mean) {

  P_1=(P_leach2+F_low)
  P_2=(P_leach2+F_high)
  P_3=(P_leach2+F_mean)

  P_all<-tibble::tibble(P_1,P_2,P_3) #table w three columns

  P_summary=rowMeans(P_all) #makes one column

  mean_P=mean(P_summary,na.rm=TRUE)
  low_P=mean(P_summary,na.rm=TRUE)-sd(P_summary,na.rm=TRUE)
  high_P=mean(P_summary,na.rm=TRUE)+sd(P_summary,na.rm=TRUE)

  x<-c(mean_P,low_P,high_P)

  return(x)

  }

#set up dfs
Soils_nested<-Soils_nested #start w df set up after running MM
sand_leach<-sand_leach%>%
  mutate(Soil_Type="sand")
loam_leach<-loam_leach%>%
  mutate(Soil_Type="loam")
clay_leach<-clay_leach%>%
  mutate(Soil_type="clay")

sand_leach_long<-sand_leach%>%
  dplyr::select(-c(contains("sd"),Soil_Type))%>% #removes the sd columns
  dplyr::select(-c(d18O_mean,d18O_mean1))%>%
  pivot_longer(cols=!c("Brand","Year"),
             names_to="stat",values_to="d18Oleach")
sand_leach_long<-sand_leach_long%>%
  mutate(scenario=case_when(
    endsWith(stat,"a") ~ "a",
    endsWith(stat,"b")~"b",
    endsWith(stat,"c")~"c",
    endsWith(stat,"d")~"d"
  ))%>%
  mutate(Soil_Type="sand")
clay_leach_long<-clay_leach%>%
```

```
  dplyr::select(-c(contains("sd"),Soil_type))%>% #removes the sd columns
  dplyr::select(-c(d18O_mean,d18O_mean1))%>%
  pivot_longer(cols=!c("Brand","Year"),
             names_to="stat",values_to="d18Oleach")
clay_leach_long<-clay_leach_long%>%
  mutate(scenario=case_when(
    endsWith(stat,"a") ~ "a",
    endsWith(stat,"b")~"b",
    endsWith(stat,"c")~"c",
    endsWith(stat,"d")~"d"
  ))%>%
  mutate(Soil_Type="clay")
loam_leach_long<-loam_leach%>%
  dplyr::select(-c(contains("sd"),Soil_Type))%>% #removes the sd columns
  dplyr::select(-c(d18O_mean,d18O_mean1))%>%
  pivot_longer(cols=!c("Brand","Year"),
             names_to="stat",values_to="d18Oleach")
loam_leach_long<-loam_leach_long%>%
  mutate(scenario=case_when(
    endsWith(stat,"a") ~ "a",
    endsWith(stat,"b")~"b",
    endsWith(stat,"c")~"c",
    endsWith(stat,"d")~"d"
  ))%>%
  mutate(Soil_Type="loam")


#separate df into a,c v b,d
Soils_nested<-Soils_nested%>%
  mutate(P_a=pmap(list(P_leach1,F_low,F_high,F_mean),P_a_formula))%>%
  mutate(P_b=pmap(list(P_leach2,F_low,F_high,F_mean),P_b_formula))

#create nested dfs based on fertiliser and scenario
sand_nested<-sand_leach_long%>%
  dplyr::group_by(Soil_Type,Brand,scenario)%>%
  nest()
clay_nested<-clay_leach_long%>%
  dplyr::group_by(Soil_Type,Brand,scenario)%>%
  nest()
loam_nested<-loam_leach_long%>%
  dplyr::group_by(Soil_Type,Brand,scenario)%>%
  nest()

#add in P content info
sand_nested<-sand_nested%>%
  dplyr::mutate(P_conc=case_when(
    scenario%in%c("a","c")~(Soils_nested[[1,17]]),
    scenario%in%c("b","d")~Soils_nested[[1,16]]
  ))
```

```
loam_nested<-loam_nested%>%
  dplyr::mutate(P_conc=case_when(
    scenario%in%c("a","c")~(Soils_nested[[2,17]]),
    scenario%in%c("b","d")~Soils_nested[[2,16]]
  ))

clay_nested<-clay_nested%>%
  dplyr::mutate(P_conc=case_when(
    scenario%in%c("a","c")~(Soils_nested[[3,17]]),
    scenario%in%c("b","d")~Soils_nested[[3,16]]
  ))


#combine dfs
nested_all<-rbind(sand_nested,loam_nested,clay_nested)

#add in spatial info
nested_all<-nested_all%>%
  mutate(area_pinjarra=case_when(
    Soil_Type=="sand"~pinjarra_sand,
    Soil_Type=="clay"~pinjarra_clay,
    Soil_Type=="loam"~pinjarra_loam
  ))%>%
  mutate(area_harvey=case_when(
    Soil_Type=="sand"~harvey_sand,
    Soil_Type=="clay"~harvey_clay,
    Soil_Type=="loam"~harvey_loam
  ))


#make up-scaling formula
#d18O_catchment=(d18Osand*f_sand*P_sand+d18Oclay*f_clay*P_clay+d18Oloam*f_loam*P_lo
am)/(P_sand+P_clay+P_loam)
nested_all<-nested_all%>%
  mutate(d18Oleach=map(data,~.x$d18Oleach))


#separate out by scenarios
nested_ac<-nested_all%>%
  filter(scenario%in%c("a","c"))
nested_bd<-nested_all%>%
  filter(scenario%in%c("b","d"))



#make formula to calculate f based on area and P conc
totals_fun<-function(df,scen) {


  #calculate total amount of P leached for Pinjarra & Harvey
```

```
totals_H<-df%>%
  dplyr::select(Soil_Type,P_conc,area_harvey,area_pinjarra)%>%
  unnest(cols=c(P_conc,area_harvey,area_pinjarra))%>%
  mutate(Phos_H=P_conc*area_harvey,
         Phos_P=P_conc*area_pinjarra)%>%
  filter(Brand=="AG"&scenario==scen)%>% #keep it simple
  ungroup()%>%
  dplyr::select(Soil_Type,Phos_H)%>%
  mutate(x=c(1,2,3,1,2,3,1,2,3))%>%
  pivot_wider(names_from=Soil_Type,values_from=Phos_H)%>%
  rowwise()%>%
  mutate(P_total=sum(across(c(sand,loam,clay))))

Hf_sand_high<-totals_H[[3,2]]/totals_H[[3,5]]
Hf_sand_low<-totals_H[[2,2]]/totals_H[[2,5]]
Hf_sand_avg<-totals_H[[1,2]]/totals_H[[1,5]]

Hf_loam_high<-totals_H[[3,3]]/totals_H[[3,5]]
Hf_loam_low<-totals_H[[2,3]]/totals_H[[2,5]]
Hf_loam_avg<-totals_H[[1,3]]/totals_H[[1,5]]

Hf_clay_high<-totals_H[[3,4]]/totals_H[[3,5]]
Hf_clay_low<-totals_H[[2,4]]/totals_H[[2,5]]
Hf_clay_avg<-totals_H[[1,4]]/totals_H[[1,5]]


H_f<-tibble::tibble(H_high=c(Hf_sand_high,Hf_loam_high,Hf_clay_high),
                    H_low=c(Hf_sand_low,Hf_loam_low,Hf_clay_low),
                    H_avg=c(Hf_sand_avg,Hf_loam_avg,Hf_clay_avg))

totals_P<-df%>%
  dplyr::select(Soil_Type,P_conc,area_harvey,area_pinjarra)%>%
  unnest(cols=c(P_conc,area_harvey,area_pinjarra))%>%
  mutate(Phos_H=P_conc*area_harvey,
         Phos_P=P_conc*area_pinjarra)%>%
  filter(Brand=="AG"&scenario==scen)%>% #keep it simple
  ungroup()%>%
  dplyr::select(Soil_Type,Phos_P)%>%
  mutate(x=c(1,2,3,1,2,3,1,2,3))%>%
  pivot_wider(names_from=Soil_Type,values_from=Phos_P)%>%
  rowwise()%>%
  mutate(P_total=sum(across(c(sand,loam,clay))))

Pf_sand_high<-totals_P[[3,2]]/totals_P[[3,5]]
Pf_sand_low<-totals_P[[2,2]]/totals_P[[2,5]]
Pf_sand_avg<-totals_P[[1,2]]/totals_P[[1,5]]

Pf_loam_high<-totals_P[[3,3]]/totals_P[[3,5]]
Pf_loam_low<-totals_P[[2,3]]/totals_P[[2,5]]
Pf_loam_avg<-totals_P[[1,3]]/totals_P[[1,5]]
```

```
  Pf_clay_high<-totals_P[[3,4]]/totals_P[[3,5]]
  Pf_clay_low<-totals_P[[2,4]]/totals_P[[2,5]]
  Pf_clay_avg<-totals_P[[1,4]]/totals_P[[1,5]]


  P_f<-tibble::tibble(P_high=c(Pf_sand_high,Pf_loam_high,Pf_clay_high),
                  P_low=c(Pf_sand_low,Pf_loam_low,Pf_clay_low),
                  P_avg=c(Pf_sand_avg,Pf_loam_avg,Pf_clay_avg))



  output<-cbind(Soil_Type=c("sand","loam","clay"),H_f,P_f)

  return(as.data.frame(output))

  }



#merge left so soil textures match up
nested_ac<-left_join(nested_ac,totals_fun(nested_ac,"a"),by="Soil_Type")

nested_bd<-left_join(nested_bd,totals_fun(nested_bd,"b"),by="Soil_Type")





#then a d18O_catchment calculation

d18O_weight_fun<-function(d18Oleach,P_low,P_high,P_avg,H_low,H_high,H_avg) {

  d18O<-tibble(x=d18Oleach)

  P_low<-expand.grid(d18O$x,P_low)%>%
    mutate(d18O_f=Var1*Var2)%>%
    summarise(
      mean=mean(d18O_f,na.rm=TRUE)
      #low=mean(d18O_f,na.rm=TRUE)-sd(d18O_f,na.rm=TRUE),
      #high=mean(d18O_f,na.rm=TRUE)+sd(d18O_f,na.rm=TRUE)
    )

  P_high<-expand.grid(d18O$x,P_high)%>%
    mutate(d18O_f=Var1*Var2)%>%
    summarise(
      mean=mean(d18O_f,na.rm=TRUE)
      #low=mean(d18O_f,na.rm=TRUE)-sd(d18O_f,na.rm=TRUE),
      #high=mean(d18O_f,na.rm=TRUE)+sd(d18O_f,na.rm=TRUE)
```

```
  )

 P_avg<-expand.grid(d18O$x,P_avg)%>%
  mutate(d18O_f=Var1*Var2)%>%
  summarise(
   mean=mean(d18O_f,na.rm=TRUE)
   #low=mean(d18O_f,na.rm=TRUE)-sd(d18O_f,na.rm=TRUE),
   #high=mean(d18O_f,na.rm=TRUE)+sd(d18O_f,na.rm=TRUE)
  )

 H_low<-expand.grid(d18O$x,H_low)%>%
  mutate(d18O_f=Var1*Var2)%>%
  summarise(
   mean=mean(d18O_f,na.rm=TRUE)
   #low=mean(d18O_f,na.rm=TRUE)-sd(d18O_f,na.rm=TRUE),
   #high=mean(d18O_f,na.rm=TRUE)+sd(d18O_f,na.rm=TRUE)
  )

 H_high<-expand.grid(d18O$x,H_high)%>%
  mutate(d18O_f=Var1*Var2)%>%
  summarise(
   mean=mean(d18O_f,na.rm=TRUE)
   #low=mean(d18O_f,na.rm=TRUE)-sd(d18O_f,na.rm=TRUE),
   #high=mean(d18O_f,na.rm=TRUE)+sd(d18O_f,na.rm=TRUE)
  )

 H_avg<-expand.grid(d18O$x,H_avg)%>%
  mutate(d18O_f=Var1*Var2)%>%
  summarise(
   mean=mean(d18O_f,na.rm=TRUE)
   #low=mean(d18O_f,na.rm=TRUE)-sd(d18O_f,na.rm=TRUE),
   #high=mean(d18O_f,na.rm=TRUE)+sd(d18O_f,na.rm=TRUE)
  )

 x<-tibble::tibble(P_frac=c("low","high","avg"),
          P=c(P_low,P_high,P_avg),
          H=c(H_low,H_high,H_avg))
 return(x)

 }

#calc for low range
d18O_weight_fun2<-function(d18Oleach,P_low,P_high,P_avg,H_low,H_high,H_avg) {

 d18O<-tibble(x=d18Oleach)

 P_low<-expand.grid(d18O$x,P_low)%>%
  mutate(d18O_f=Var1*Var2)%>%
  summarise(
   low=mean(d18O_f,na.rm=TRUE)-sd(d18O_f,na.rm=TRUE)
```

```
    #high=mean(d18O_f,na.rm=TRUE)+sd(d18O_f,na.rm=TRUE)
  )

 P_high<-expand.grid(d18O$x,P_high)%>%
  mutate(d18O_f=Var1*Var2)%>%
  summarise(
    #mean=mean(d18O_f,na.rm=TRUE)
    low=mean(d18O_f,na.rm=TRUE)-sd(d18O_f,na.rm=TRUE)
    #high=mean(d18O_f,na.rm=TRUE)+sd(d18O_f,na.rm=TRUE)
  )

 P_avg<-expand.grid(d18O$x,P_avg)%>%
  mutate(d18O_f=Var1*Var2)%>%
  summarise(
    #mean=mean(d18O_f,na.rm=TRUE)
    low=mean(d18O_f,na.rm=TRUE)-sd(d18O_f,na.rm=TRUE)
    #high=mean(d18O_f,na.rm=TRUE)+sd(d18O_f,na.rm=TRUE)
  )

 H_low<-expand.grid(d18O$x,H_low)%>%
  mutate(d18O_f=Var1*Var2)%>%
  summarise(
    #mean=mean(d18O_f,na.rm=TRUE)
    low=mean(d18O_f,na.rm=TRUE)-sd(d18O_f,na.rm=TRUE)
    #high=mean(d18O_f,na.rm=TRUE)+sd(d18O_f,na.rm=TRUE)
  )

 H_high<-expand.grid(d18O$x,H_high)%>%
  mutate(d18O_f=Var1*Var2)%>%
  summarise(
    #mean=mean(d18O_f,na.rm=TRUE)
    low=mean(d18O_f,na.rm=TRUE)-sd(d18O_f,na.rm=TRUE)
    #high=mean(d18O_f,na.rm=TRUE)+sd(d18O_f,na.rm=TRUE)
  )

 H_avg<-expand.grid(d18O$x,H_avg)%>%
  mutate(d18O_f=Var1*Var2)%>%
  summarise(
    #mean=mean(d18O_f,na.rm=TRUE)
    low=mean(d18O_f,na.rm=TRUE)-sd(d18O_f,na.rm=TRUE)
    #high=mean(d18O_f,na.rm=TRUE)+sd(d18O_f,na.rm=TRUE)
  )

 x<-tibble::tibble(P_frac=c("low","high","avg"),
                   P=c(P_low,P_high,P_avg),
                   H=c(H_low,H_high,H_avg))
 return(x)

}
```

```
#calc for high range
d18O_weight_fun3<-function(d18Oleach,P_low,P_high,P_avg,H_low,H_high,H_avg) {

  d18O<-tibble(x=d18Oleach)

  P_low<-expand.grid(d18O$x,P_low)%>%
    mutate(d18O_f=Var1*Var2)%>%
    summarise(
      #low=mean(d18O_f,na.rm=TRUE)-sd(d18O_f,na.rm=TRUE)
      high=mean(d18O_f,na.rm=TRUE)+sd(d18O_f,na.rm=TRUE)
    )

  P_high<-expand.grid(d18O$x,P_high)%>%
    mutate(d18O_f=Var1*Var2)%>%
    summarise(
      #mean=mean(d18O_f,na.rm=TRUE)
      #low=mean(d18O_f,na.rm=TRUE)-sd(d18O_f,na.rm=TRUE)
      high=mean(d18O_f,na.rm=TRUE)+sd(d18O_f,na.rm=TRUE)
    )

  P_avg<-expand.grid(d18O$x,P_avg)%>%
    mutate(d18O_f=Var1*Var2)%>%
    summarise(
      #mean=mean(d18O_f,na.rm=TRUE)
      #low=mean(d18O_f,na.rm=TRUE)-sd(d18O_f,na.rm=TRUE)
      high=mean(d18O_f,na.rm=TRUE)+sd(d18O_f,na.rm=TRUE)
    )

  H_low<-expand.grid(d18O$x,H_low)%>%
    mutate(d18O_f=Var1*Var2)%>%
    summarise(
      #mean=mean(d18O_f,na.rm=TRUE)
      #low=mean(d18O_f,na.rm=TRUE)-sd(d18O_f,na.rm=TRUE)
      high=mean(d18O_f,na.rm=TRUE)+sd(d18O_f,na.rm=TRUE)
    )

  H_high<-expand.grid(d18O$x,H_high)%>%
    mutate(d18O_f=Var1*Var2)%>%
    summarise(
      #mean=mean(d18O_f,na.rm=TRUE)
      #low=mean(d18O_f,na.rm=TRUE)-sd(d18O_f,na.rm=TRUE)
      high=mean(d18O_f,na.rm=TRUE)+sd(d18O_f,na.rm=TRUE)
    )

  H_avg<-expand.grid(d18O$x,H_avg)%>%
    mutate(d18O_f=Var1*Var2)%>%
    summarise(
      #mean=mean(d18O_f,na.rm=TRUE)
      #low=mean(d18O_f,na.rm=TRUE)-sd(d18O_f,na.rm=TRUE)
      high=mean(d18O_f,na.rm=TRUE)+sd(d18O_f,na.rm=TRUE)
```

```
  )

  x<-tibble::tibble(P_frac=c("low","high","avg"),
                P=c(P_low,P_high,P_avg),
                H=c(H_low,H_high,H_avg))
  return(x)

}

nested_ac<-nested_ac%>%
  mutate(d18Op=pmap(list(d18Oleach,P_low,P_high,P_avg,H_low,H_high,H_avg),
                  d18O_weight_fun))
nested_bd<-nested_bd%>%
  mutate(d18Op=pmap(list(d18Oleach,P_low,P_high,P_avg,H_low,H_high,H_avg),
                d18O_weight_fun))

nested_ac2<-nested_ac%>%
  mutate(d18Op=pmap(list(d18Oleach,P_low,P_high,P_avg,H_low,H_high,H_avg),
                d18O_weight_fun2))
nested_bd2<-nested_bd%>%
  mutate(d18Op=pmap(list(d18Oleach,P_low,P_high,P_avg,H_low,H_high,H_avg),
                d18O_weight_fun2))
nested_ac3<-nested_ac%>%
  mutate(d18Op=pmap(list(d18Oleach,P_low,P_high,P_avg,H_low,H_high,H_avg),
                d18O_weight_fun3))
nested_bd3<-nested_bd%>%
  mutate(d18Op=pmap(list(d18Oleach,P_low,P_high,P_avg,H_low,H_high,H_avg),
                d18O_weight_fun3))


#so now add together some mixes, so:

#mix 1: 90% SP, 20%c + 80%d
#mix 2: 90% SP, 50%c + 50%d
#mix 3: 90% SP, 5%c + 95%d

#need outputs for both Harvey v Pinjarra
#make df that just has d18O and f_area info in it
upscaled_d18O_df<-rbind(nested_ac%>%
                    dplyr::select(area_pinjarra,area_harvey,d18Op)%>%
                    unnest(cols=c(d18Op),names_sep="_"),
                  nested_bd%>%
                    dplyr::select(area_pinjarra,area_harvey,d18Op)%>%
                    unnest(cols=c(d18Op),names_sep="_")
)%>%
  rename(frac=d18Op_P_frac)

#convert columns from lists to numeric
upscaled_d18O_df$d18Op_H<-as.numeric(upscaled_d18O_df$d18Op_H)
upscaled_d18O_df$d18Op_P<-as.numeric(upscaled_d18O_df$d18Op_P)
```

```
#for low range
upscaled_d18O_df2<-rbind(nested_ac2%>%
                         dplyr::select(area_pinjarra,area_harvey,d18Op)%>%
                         unnest(cols=c(d18Op),names_sep="_"),
                       nested_bd2%>%
                         dplyr::select(area_pinjarra,area_harvey,d18Op)%>%
                         unnest(cols=c(d18Op),names_sep="_")
)%>%
  rename(frac=d18Op_P_frac)

#convert columns from lists to numeric
upscaled_d18O_df2$d18Op_H<-as.numeric(upscaled_d18O_df2$d18Op_H)
upscaled_d18O_df2$d18Op_P<-as.numeric(upscaled_d18O_df2$d18Op_P)

#for high range
upscaled_d18O_df3<-rbind(nested_ac3%>%
                         dplyr::select(area_pinjarra,area_harvey,d18Op)%>%
                         unnest(cols=c(d18Op),names_sep="_"),
                       nested_bd3%>%
                         dplyr::select(area_pinjarra,area_harvey,d18Op)%>%
                         unnest(cols=c(d18Op),names_sep="_")
)%>%
  rename(frac=d18Op_P_frac)

#convert columns from lists to numeric
upscaled_d18O_df3$d18Op_H<-as.numeric(upscaled_d18O_df3$d18Op_H)
upscaled_d18O_df3$d18Op_P<-as.numeric(upscaled_d18O_df3$d18Op_P)

#function to play around with mixing
mixing_fun<-function(df,f_SP,f_a) {

  #define terms
  f_AG=1-f_SP #so either SP or AG fertilisers
  f_d=1-f_a #either scenario a or d

{P_avg<-(f_SP*(f_a*(as.double(df%>%ungroup()%>%
                         filter(Brand=="SP",scenario=="a",
                                Soil_Type=="sand",frac=="avg")%>%
                         dplyr::select(d18Op_P))+
                as.double(df%>%ungroup()%>%
                          filter(Brand=="SP",scenario=="a",
                                 Soil_Type=="clay",frac=="avg")%>%
                          dplyr::select(d18Op_P))+
                as.double(df%>%ungroup()%>%
                          filter(Brand=="SP",scenario=="a",
                                 Soil_Type=="loam",frac=="avg")%>%
                          dplyr::select(d18Op_P))
)+f_d*(as.double(df%>%ungroup()%>%
                  filter(Brand=="SP",scenario=="d",
```

```
                    Soil_Type=="sand",frac=="avg")%>%
                dplyr::select(d18Op_P))+
        as.double(df%>%ungroup()%>%
                filter(Brand=="SP",scenario=="d",
                    Soil_Type=="clay",frac=="avg")%>%
                dplyr::select(d18Op_P))+
        as.double(df%>%ungroup()%>%
                filter(Brand=="SP",scenario=="d",
                    Soil_Type=="loam",frac=="avg")%>%
                dplyr::select(d18Op_P))
)
)+f_AG*(f_a*(as.double(df%>%ungroup()%>%
                    filter(Brand=="AG",scenario=="a",
                        Soil_Type=="sand",frac=="avg")%>%
                    dplyr::select(d18Op_P))+
            as.double(df%>%ungroup()%>%
                    filter(Brand=="AG",scenario=="a",
                        frac=="avg",Soil_Type=="clay")%>%
                    dplyr::select(d18Op_P))+
            as.double(df%>%ungroup()%>%
                    filter(Brand=="AG",scenario=="a",
                        Soil_Type=="loam",frac=="avg")%>%
                    dplyr::select(d18Op_P))
)+f_d*(as.double(df%>%ungroup()%>%
                filter(Brand=="AG",scenario=="d",
                    Soil_Type=="sand",frac=="avg")%>%
                dplyr::select(d18Op_P))+
        as.double(df%>%ungroup()%>%
                filter(Brand=="AG",scenario=="d",
                    Soil_Type=="clay",frac=="avg")%>%
                dplyr::select(d18Op_P))+
        as.double(df%>%ungroup()%>%
                filter(Brand=="AG",scenario=="d",
                    Soil_Type=="loam",frac=="avg")%>%
                dplyr::select(d18Op_P))
)
))

P_low<-(f_SP*(f_a*(as.double(df%>%ungroup()%>%
                        filter(Brand=="SP",scenario=="a",
                            Soil_Type=="sand",frac=="low")%>%
                        dplyr::select(d18Op_P))+
                as.double(df%>%ungroup()%>%
                        filter(Brand=="SP",scenario=="a",
                            frac=="low",Soil_Type=="clay")%>%
                        dplyr::select(d18Op_P))+
                as.double(df%>%ungroup()%>%
                        filter(Brand=="SP",scenario=="a",
                            frac=="low",Soil_Type=="loam")%>%
                        dplyr::select(d18Op_P))
```

```
)+f_d*(as.double(df%>%ungroup()%>%
            filter(Brand=="SP",scenario=="d",
                frac=="low",Soil_Type=="sand")%>%
            dplyr::select(d18Op_P))+
      as.double(df%>%ungroup()%>%
             filter(Brand=="SP",scenario=="d",
                 frac=="low",Soil_Type=="clay")%>%
             dplyr::select(d18Op_P))+
      as.double(df%>%ungroup()%>%
             filter(Brand=="SP",scenario=="d",
                 frac=="low",Soil_Type=="loam")%>%
             dplyr::select(d18Op_P))
)
)+f_AG*(f_a*(as.double(df%>%ungroup()%>%
                filter(Brand=="AG",scenario=="a",
                    frac=="low",Soil_Type=="sand")%>%
                dplyr::select(d18Op_P))+
          as.double(df%>%ungroup()%>%
                 filter(Brand=="AG",scenario=="a",
                     frac=="low",Soil_Type=="clay")%>%
                 dplyr::select(d18Op_P))+
          as.double(df%>%ungroup()%>%
                 filter(Brand=="AG",scenario=="a",
                     frac=="low",Soil_Type=="loam")%>%
                 dplyr::select(d18Op_P))
)+f_d*(as.double(df%>%ungroup()%>%
            filter(Brand=="AG",scenario=="d",
                frac=="low",Soil_Type=="sand")%>%
            dplyr::select(d18Op_P))+
      as.double(df%>%ungroup()%>%
             filter(Brand=="AG",scenario=="d",
                 frac=="low",Soil_Type=="clay")%>%
             dplyr::select(d18Op_P))+
      as.double(df%>%ungroup()%>%
             filter(Brand=="AG",scenario=="d",
                 frac=="low",Soil_Type=="loam")%>%
             dplyr::select(d18Op_P))
)
))

P_high<-(f_SP*(f_a*(as.double(df%>%ungroup()%>%
                    filter(Brand=="SP",scenario=="a",
                        frac=="high",Soil_Type=="sand")%>%
                    dplyr::select(d18Op_P))+
              as.double(df%>%ungroup()%>%
                     filter(Brand=="SP",scenario=="a",
                         frac=="high",Soil_Type=="clay")%>%
                     dplyr::select(d18Op_P))+
              as.double(df%>%ungroup()%>%
                     filter(Brand=="SP",scenario=="a",
```

```
                                frac=="high",Soil_Type=="loam")%>%
                            dplyr::select(d18Op_P))
)+f_d*(as.double(df%>%ungroup()%>%
               filter(Brand=="SP",scenario=="d",
                   frac=="high",Soil_Type=="sand")%>%
               dplyr::select(d18Op_P))+
        as.double(df%>%ungroup()%>%
                filter(Brand=="SP",scenario=="d",
                    frac=="high",Soil_Type=="clay")%>%
                dplyr::select(d18Op_P))+
        as.double(df%>%ungroup()%>%
                filter(Brand=="SP",scenario=="d",
                    frac=="high",Soil_Type=="loam")%>%
                dplyr::select(d18Op_P))
)
)+f_AG*(f_a*(as.double(df%>%ungroup()%>%
                    filter(Brand=="AG",scenario=="a",
                        frac=="high",Soil_Type=="sand")%>%
                    dplyr::select(d18Op_P))+
             as.double(df%>%ungroup()%>%
                     filter(Brand=="AG",scenario=="a",
                         frac=="high",Soil_Type=="clay")%>%
                     dplyr::select(d18Op_P))+
             as.double(df%>%ungroup()%>%
                     filter(Brand=="AG",scenario=="a",
                         frac=="high",Soil_Type=="loam")%>%
                     dplyr::select(d18Op_P))
)+f_d*(as.double(df%>%ungroup()%>%
               filter(Brand=="AG",scenario=="d",
                   frac=="high",Soil_Type=="sand")%>%
               dplyr::select(d18Op_P))+
        as.double(df%>%ungroup()%>%
                filter(Brand=="AG",scenario=="d",
                    frac=="high",Soil_Type=="clay")%>%
                dplyr::select(d18Op_P))+
        as.double(df%>%ungroup()%>%
                filter(Brand=="AG",scenario=="d",
                    frac=="high",Soil_Type=="loam")%>%
                dplyr::select(d18Op_P))
)
))
}
 #harvey calcs
 H_avg<-(f_SP*(f_a*(as.double(df%>%ungroup()%>%
                            filter(Brand=="SP",scenario=="a",
                                frac=="avg",Soil_Type=="sand")%>%
                            dplyr::select(d18Op_H))+
                     as.double(df%>%ungroup()%>%
                             filter(Brand=="SP",scenario=="a",
                                 frac=="avg",Soil_Type=="clay")%>%
```

```
                        dplyr::select(d18Op_H))+
                  as.double(df%>%ungroup()%>%
                        filter(Brand=="SP",scenario=="a",
                           frac=="avg",Soil_Type=="loam")%>%
                        dplyr::select(d18Op_H))
)+f_d*(as.double(df%>%ungroup()%>%
                filter(Brand=="SP",scenario=="d",
                   frac=="avg",Soil_Type=="sand")%>%
                dplyr::select(d18Op_H))+
        as.double(df%>%ungroup()%>%
                 filter(Brand=="SP",scenario=="d",
                    frac=="avg",Soil_Type=="clay")%>%
                 dplyr::select(d18Op_H))+
        as.double(df%>%ungroup()%>%
                 filter(Brand=="SP",scenario=="d",
                    frac=="avg",Soil_Type=="loam")%>%
                 dplyr::select(d18Op_H))
)
)+f_AG*(f_a*(as.double(df%>%ungroup()%>%
                   filter(Brand=="AG",scenario=="a",
                      frac=="avg",Soil_Type=="sand")%>%
                   dplyr::select(d18Op_H))+
             as.double(df%>%ungroup()%>%
                    filter(Brand=="AG",scenario=="a",
                       frac=="avg",Soil_Type=="clay")%>%
                    dplyr::select(d18Op_H))+
             as.double(df%>%ungroup()%>%
                    filter(Brand=="AG",scenario=="a",
                       frac=="avg",Soil_Type=="loam")%>%
                    dplyr::select(d18Op_H))
)+f_d*(as.double(df%>%ungroup()%>%
                filter(Brand=="AG",scenario=="d",
                   frac=="avg",Soil_Type=="sand")%>%
                dplyr::select(d18Op_P))+
        as.double(df%>%ungroup()%>%
                 filter(Brand=="AG",scenario=="d",
                    frac=="avg",Soil_Type=="clay")%>%
                 dplyr::select(d18Op_H))+
        as.double(df%>%ungroup()%>%
                 filter(Brand=="AG",scenario=="d",
                    frac=="avg",Soil_Type=="loam")%>%
                 dplyr::select(d18Op_H))
)
))

H_low<-(f_SP*(f_a*(as.double(df%>%ungroup()%>%
                      filter(Brand=="SP",scenario=="a",
                         frac=="low",Soil_Type=="sand")%>%
                      dplyr::select(d18Op_H))+
                 as.double(df%>%ungroup()%>%
```

```
                    filter(Brand=="SP",scenario=="a",
                        frac=="low",Soil_Type=="clay")%>%
                    dplyr::select(d18Op_H))+
                as.double(df%>%ungroup()%>%
                    filter(Brand=="SP",scenario=="a",
                        frac=="low",Soil_Type=="loam")%>%
                    dplyr::select(d18Op_H))
)+f_d*(as.double(df%>%ungroup()%>%
            filter(Brand=="SP",scenario=="d",
                frac=="low",Soil_Type=="sand")%>%
            dplyr::select(d18Op_H))+
        as.double(df%>%ungroup()%>%
            filter(Brand=="SP",scenario=="d",
                frac=="low",Soil_Type=="clay")%>%
            dplyr::select(d18Op_H))+
        as.double(df%>%ungroup()%>%
            filter(Brand=="SP",scenario=="d",
                frac=="low",Soil_Type=="loam")%>%
            dplyr::select(d18Op_H))
)
)+f_AG*(f_a*(as.double(df%>%ungroup()%>%
                filter(Brand=="AG",scenario=="a",
                    frac=="low",Soil_Type=="sand")%>%
                dplyr::select(d18Op_H))+
            as.double(df%>%ungroup()%>%
                filter(Brand=="AG",scenario=="a",
                    frac=="low",Soil_Type=="clay")%>%
                dplyr::select(d18Op_H))+
            as.double(df%>%ungroup()%>%
                filter(Brand=="AG",scenario=="a",
                    frac=="low",Soil_Type=="loam")%>%
                dplyr::select(d18Op_H))
)+f_d*(as.double(df%>%ungroup()%>%
            filter(Brand=="AG",scenario=="d",
                frac=="low",Soil_Type=="sand")%>%
            dplyr::select(d18Op_H))+
        as.double(df%>%ungroup()%>%
            filter(Brand=="AG",scenario=="d",
                frac=="low",Soil_Type=="clay")%>%
            dplyr::select(d18Op_H))+
        as.double(df%>%ungroup()%>%
            filter(Brand=="AG",scenario=="d",
                frac=="low",Soil_Type=="loam")%>%
            dplyr::select(d18Op_H))
)
))

H_high<-(f_SP*(f_a*(as.double(df%>%ungroup()%>%
                    filter(Brand=="SP",scenario=="a",
                        frac=="high",Soil_Type=="sand")%>%
```

```
                  dplyr::select(d18Op_H))+
              as.double(df%>%ungroup()%>%
                    filter(Brand=="SP",scenario=="a",
                        frac=="high",Soil_Type=="clay")%>%
                    dplyr::select(d18Op_H))+
              as.double(df%>%ungroup()%>%
                    filter(Brand=="SP",scenario=="a",
                        frac=="high",Soil_Type=="loam")%>%
                    dplyr::select(d18Op_H))
  )+f_d*(as.double(df%>%ungroup()%>%
            filter(Brand=="SP",scenario=="d",
                frac=="high",Soil_Type=="sand")%>%
            dplyr::select(d18Op_H))+
      as.double(df%>%ungroup()%>%
             filter(Brand=="SP",scenario=="d",
                 frac=="high",Soil_Type=="clay")%>%
             dplyr::select(d18Op_H))+
      as.double(df%>%ungroup()%>%
             filter(Brand=="SP",scenario=="d",
                 frac=="high",Soil_Type=="loam")%>%
             dplyr::select(d18Op_H))
  )
  )+f_AG*(f_a*(as.double(df%>%ungroup()%>%
                filter(Brand=="AG",scenario=="a",
                    frac=="high",Soil_Type=="sand")%>%
                dplyr::select(d18Op_H))+
         as.double(df%>%ungroup()%>%
                 filter(Brand=="AG",scenario=="a",
                     frac=="high",Soil_Type=="clay")%>%
                 dplyr::select(d18Op_H))+
         as.double(df%>%ungroup()%>%
                 filter(Brand=="AG",scenario=="a",
                     frac=="high",Soil_Type=="loam")%>%
                 dplyr::select(d18Op_H))
  )+f_d*(as.double(df%>%ungroup()%>%
            filter(Brand=="AG",scenario=="d",
                frac=="high",Soil_Type=="sand")%>%
            dplyr::select(d18Op_H))+
      as.double(df%>%ungroup()%>%
             filter(Brand=="AG",scenario=="d",
                 frac=="high",Soil_Type=="clay")%>%
             dplyr::select(d18Op_H))+
      as.double(df%>%ungroup()%>%
             filter(Brand=="AG",scenario=="d",
                 frac=="high",Soil_Type=="loam")%>%
             dplyr::select(d18Op_H))
  )
  ))

output<-list(Pinjarra_avg=P_avg,Pinjarra_high=P_high,Pinjarra_low=P_low,
```

```
        Harvey_avg=H_avg,Harvey_low=H_low,Harvey_high=H_high)

return(output)

}

#outputs using mean d18OP(export)
upscale_table<-data.frame(rbind(S1=mixing_fun(upscaled_d18O_df,1,0.1), #range for scenario w 100% SP fertiliser, 10% scenario a
                     S2=mixing_fun(upscaled_d18O_df,1,0.5),
                     S3=mixing_fun(upscaled_d18O_df,0.6,0.1),
                     S4=mixing_fun(upscaled_d18O_df,0.6,0.5))) #range for scenario w 60% SP fertiliser, 50% scenario a
#outputs using low range d18OP(export)
upscale_table2<-data.frame(rbind(S1=mixing_fun(upscaled_d18O_df2,1,0.1),
                         S2=mixing_fun(upscaled_d18O_df2,1,0.5),
                         S3=mixing_fun(upscaled_d18O_df2,0.6,0.1),
                         S4=mixing_fun(upscaled_d18O_df2,0.6,0.5)))
#outputs using high range d18OP(export)
upscale_table3<-data.frame(rbind(S1=mixing_fun(upscaled_d18O_df2,1,0.1),
                         S2=mixing_fun(upscaled_d18O_df3,1,0.5),
                         S3=mixing_fun(upscaled_d18O_df3,0.6,0.1),
                         S4=mixing_fun(upscaled_d18O_df3,0.6,0.5)))
#Table 3 - select max-min for each scenario x sub-catchment
```

S5: References

Bünemann, E. K., Assessment of gross and net mineralization rates of soil organic phosphorus - A review. *Soil Biol. Biochem.* **2015,** *89*, 82-98.

Burkitt, L. L.; Moody, P. W.; Gourley, C. J. P.; Hannah, M. C., A simple phosphorus buffering index for Australian soils. *Soil Res.* **2002,** *40* (3), 497-513.

Helfenstein, J.; Pistocchi, C.; Oberson, A.; Tamburini, F.; Goll, D. S.; Frossard, E., Estimates of mean residence times of phosphorus in commonly considered inorganic soil phosphorus pools. *Biogeosci.* **2020,** *17* (2), 441-454.

Kearney, M. R., MicroclimOz – A microclimate data set for Australia, with example applications. *Austral Ecol.* **2019,** *44* (3), 534-544.

McArthur, W. M.; Bettenay, E., *The development and distribution of the soils of the Swan Coastal Plain, Western Australia*. 2nd ed.; CSIRO: Melbourne, Australia, 1974.

van den Hoogen, J.; Lembrechts, J. J.; Aalto, J.; Ashcroft, M. B.; De Frenne, P.; Kemppinen, J.; Kopecký, M.; Luoto, M.; Maclean, I. M. D.; Crowther, T.; Bailey, J. J.; Haesen, S.; Klinges, D. H.; Niittynen, P.; Scheffers, B. R.; Van Meerbeek, K.; SoilTempConsortium; Nijs, I.; Lenoir, J., Global Soil Temperature code and data. Zenodo, Ed. 2021.

Weller, H. countcolors: Locates and Counts Pixels Within Color Range(s) in Images. https://CRAN.R-project.org/package=countcolors.