

SGP Appendix 2: Spatial-temporal Bootstrap and Boxplots

Richard G. Stockey

March 2022

1. Filter primary dataset to generate proxy-specific subsets

First, import the primary dataset that was generated in Appendix 1.

```
load("Filtered.trace.toc.full.20230205.RData")
nrow(trace.toc.full)
```

```
## [1] 25600
```

Load required packages.

```
library(deeptime)
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.2.3
```

```
library(ggplot2)
library(ggthemes)
```

```
## Warning: package 'ggthemes' was built under R version 4.2.3
```

```
library(maps)
```

```
## Warning: package 'maps' was built under R version 4.2.3
```

```
library(pals)
library(simpleboot)
library(sp)
```

```
## Warning: package 'sp' was built under R version 4.2.3
```

Update base of the Cryogenian in deeptime.

```
periods.edit <- deeptime::periods
periods.edit[14,2] <- 720
periods.edit[15,3] <- 720
periods.edit$color[13:15] <- c("#FED96A", "#FECC5C", "#FEBF4E")
```

As this is a spatial-temporal bootstrap, we remove samples with no geospatial data.

```
trace.toc.full <- filter(trace.toc.full, !is.na(site.latitude))
trace.toc.full <- filter(trace.toc.full, !is.na(site.longitude))
nrow(trace.toc.full)
```

```
## [1] 25331
```

Note that in all subset datasets below the suffix “.box” is used to denote that these subsets are generated for spatial-temporal bootstrap analyses that are plotted as box and whisker plots. After generating each subset we establish the size of the new dataset.

Molybdenum

For Mo spatial-temporal bootstrap analyses, we are only interested in samples deposited in euxinic depositional environments. We therefore filter the primary dataset to include only samples that are classified as euxinic based upon the iron speciation proxy. We also remove any samples with no Mo data.

```
Mo.eux.box <- trace.toc.full %>%
  filter(!is.na(Mo..ppm.)) %>%
  filter(FeHR.FeT >= 0.38 & Fe.py.FeHR >= 0.7)

nrow(Mo.eux.box)
```

```
## [1] 882
```

Uranium

For U spatial-temporal bootstrap analyses, we are only interested in samples deposited in anoxic (ferruginous or euxinic) depositional environments. We therefore filter the primary dataset to include only samples that are classified as anoxic based upon iron speciation or Fe/Al ratios. We also remove any samples with no U data.

```
U.anox.box <- trace.toc.full %>%
  filter(!is.na(U..ppm.)) %>%
  filter(FeHR.FeT >= 0.38 | FeT.Al >= 0.53)

nrow(U.anox.box)
```

```
## [1] 4739
```

Proportion euxinic

For spatial-temporal bootstrap analyses of the proportion of samples that are euxinic, we are only interested in samples deposited in anoxic (ferruginous or euxinic) depositional environments. Given that we use iron speciation to determine the proportion of euxinic samples (and therefore all samples in this analyses must have full iron speciation data), we therefore filter the primary dataset to include only samples that are classified as anoxic based upon iron speciation data. We also remove samples with no FePy/FeHR data, as with the other analyses (although the requirement of the filtering step to have FeHR/FeT data should achieve this as no samples should have partial iron speciation data).

```
Fepy.anox.box <- trace.toc.full %>%
  filter(!is.na(Fe.py.FeHR)) %>%
  filter(FeHR.FeT >= 0.38)

nrow(Fepy.anox.box)
```

```
## [1] 3832
```

For the analysis of the proportion of euxinic samples, we also need to code samples based upon whether they are euxinic (based on iron speciation) in a binary fashion.

```
Fepy.anox.box$euxinic.Fe[Fepy.anox.box$Fe.py.FeHR >= 0.7] <- 1
Fepy.anox.box$euxinic.Fe[Fepy.anox.box$Fe.py.FeHR < 0.7] <- 0
```

Total organic carbon

For spatial-temporal bootstrap analyses of total organic carbon we use no redox filter. We just remove samples with no TOC data.

```
TOC.all.box <- trace.toc.full %>%
  filter(!is.na(TOC..wt..))

nrow(TOC.all.box)
```

```
## [1] 12369
```

2. Assign samples into temporal bins

Set age parameters for temporal binning.

```
bin_size <- 25
start_age <- 1000
end_age <- 300
```

Bin samples.

```

Mo.eux.box$time.bin <- seq(end_age, start_age, bin_size)[as.numeric(cut(Mo.eux.box$interpreted.age, seq(end_age, start_age, bin_size)))]+bin_size/2

U.anox.box$time.bin <- seq(end_age, start_age, bin_size)[as.numeric(cut(U.anox.box$interpreted.age, seq(end_age, start_age, bin_size)))]+bin_size/2

Fepy.anox.box$time.bin <- seq(end_age, start_age, bin_size)[as.numeric(cut(Fepy.anox.box$interpreted.age, seq(end_age, start_age, bin_size)))]+bin_size/2

TOC.all.box$time.bin <- seq(end_age, start_age, bin_size)[as.numeric(cut(TOC.all.box$interpreted.age, seq(end_age, start_age, bin_size)))]+bin_size/2

```

Check binning worked correctly. Ask for how many rows of samples it is true that the sample's interpreted age lies within the bounds of the assigned bin (middle of the bin +/- half of the bin size).

```

bin.test.Mo <- as.character()
for(row in 1:nrow(Mo.eux.box)){
  bin.test.Mo[row] <- between(Mo.eux.box$interpreted.age[row], Mo.eux.box$time.bin[row]-(bin_size/2), Mo.eux.box$time.bin[row]+(bin_size/2))
}
summary(as.factor(bin.test.Mo))

```

```

## TRUE
## 882

```

```

bin.test.U <- as.character()
for(row in 1:nrow(U.anox.box)){
  bin.test.U[row] <- between(U.anox.box$interpreted.age[row], U.anox.box$time.bin[row]-(bin_size/2), U.anox.box$time.bin[row]+(bin_size/2))
}
summary(as.factor(bin.test.U))

```

```

## TRUE
## 4739

```

```

bin.test.Fepy <- as.character()
for(row in 1:nrow(Fepy.anox.box)){
  bin.test.Fepy[row] <- between(Fepy.anox.box$interpreted.age[row], Fepy.anox.box$time.bin[row]-(bin_size/2), Fepy.anox.box$time.bin[row]+(bin_size/2))
}
summary(as.factor(bin.test.Fepy))

```

```

## TRUE
## 3832

```

```

bin.test.TOC <- as.character()
for(row in 1:nrow(TOC.all.box)){
  bin.test.TOC[row] <- between(TOC.all.box$interpreted.age[row], TOC.all.box$time.bi
n[row]-(bin_size/2), TOC.all.box$time.bin[row]+(bin_size/2))
}
summary(as.factor(bin.test.TOC))

```

```

##  TRUE
## 12369

```

It is true that the interpreted age of each sample in the following analysis lies within the age bin that it has been assigned.

3. Box and whisker plots of raw data

Generate box and whisker plots of raw data

Mo

```

Mo.box.raw.standalone <- ggplot(Mo.eux.box, aes(x=time.bin, y=Mo..ppm., group=time
.bin))+

  stat_boxplot(geom = 'errorbar', size=0.6, width = 6, color="grey20", lwd=.1) +
  geom_boxplot(outlier.alpha = .5, outlier.shape=16, outlier.size=2, outlier.color
="grey20", width=15, color="grey20", lwd=0.6, fill="grey80", fatten = 1.3)+

  coord_geo(xlim=c(start_age+1,end_age-1), ylim=c(-.5,325),expand=FALSE,
             pos = as.list(rep("bottom", 1)),
             abbrv=list( T),
             dat = list(periods.edit),
             height = list(unit(2, "lines")),
             size=list(7),
             bord=list(c("left", "bottom", "right")), lwd=as.list(1))+

  theme_minimal()+scale_x_reverse(breaks=c(300,400,500,600,700,800,900,1000))+#bre
aks=c(360,400,440,480))+

  ylab("Mo (ppm)")+xlab("Time (Ma)")+

  theme(plot.margin = ggplot2::margin(0,1,1,1,"cm"),panel.border = element_rect(fi
ll=NA,color="black", size=2,linetype="solid"),
        axis.ticks = element_line(size=1),
        axis.line = element_line(lineend = 'square'),
        axis.title = element_text(size=34),
        axis.text = element_text( size=26, color="black"),
        legend.title = element_text(size=24),
        legend.text = element_text( size=18),
        axis.ticks.length = unit(5, "points"),
        legend.position="none",
        panel.grid.major = element_blank(),panel.grid.minor = element_blank())

```

```

## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

```

```

## Warning: The `size` argument of `element_line()` is deprecated as of ggplot2 3.
4.0.
## i Please use the `linewidth` argument instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

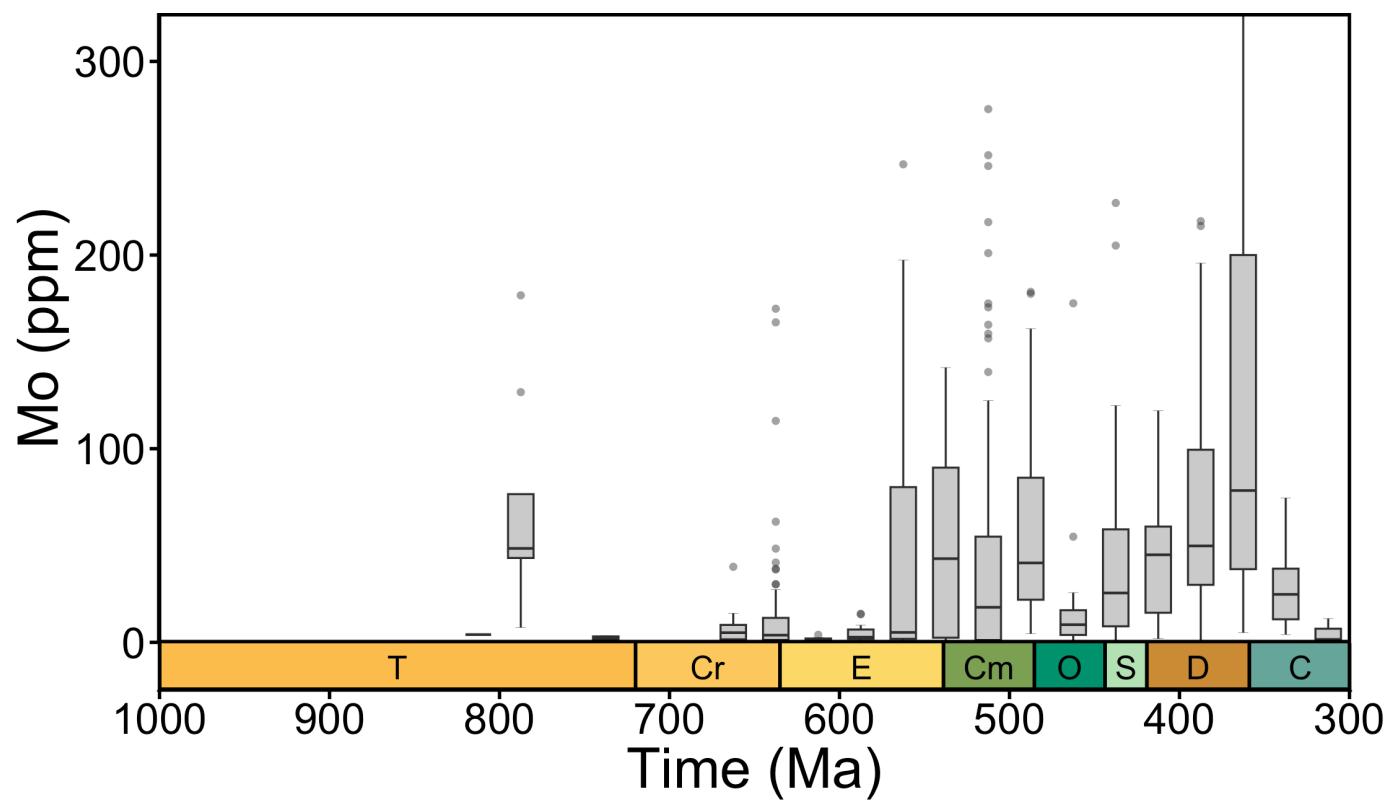
```

```

## Warning: The `size` argument of `element_rect()` is deprecated as of ggplot2 3.
4.0.
## i Please use the `linewidth` argument instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

```

Mo.box.raw.standalone



U

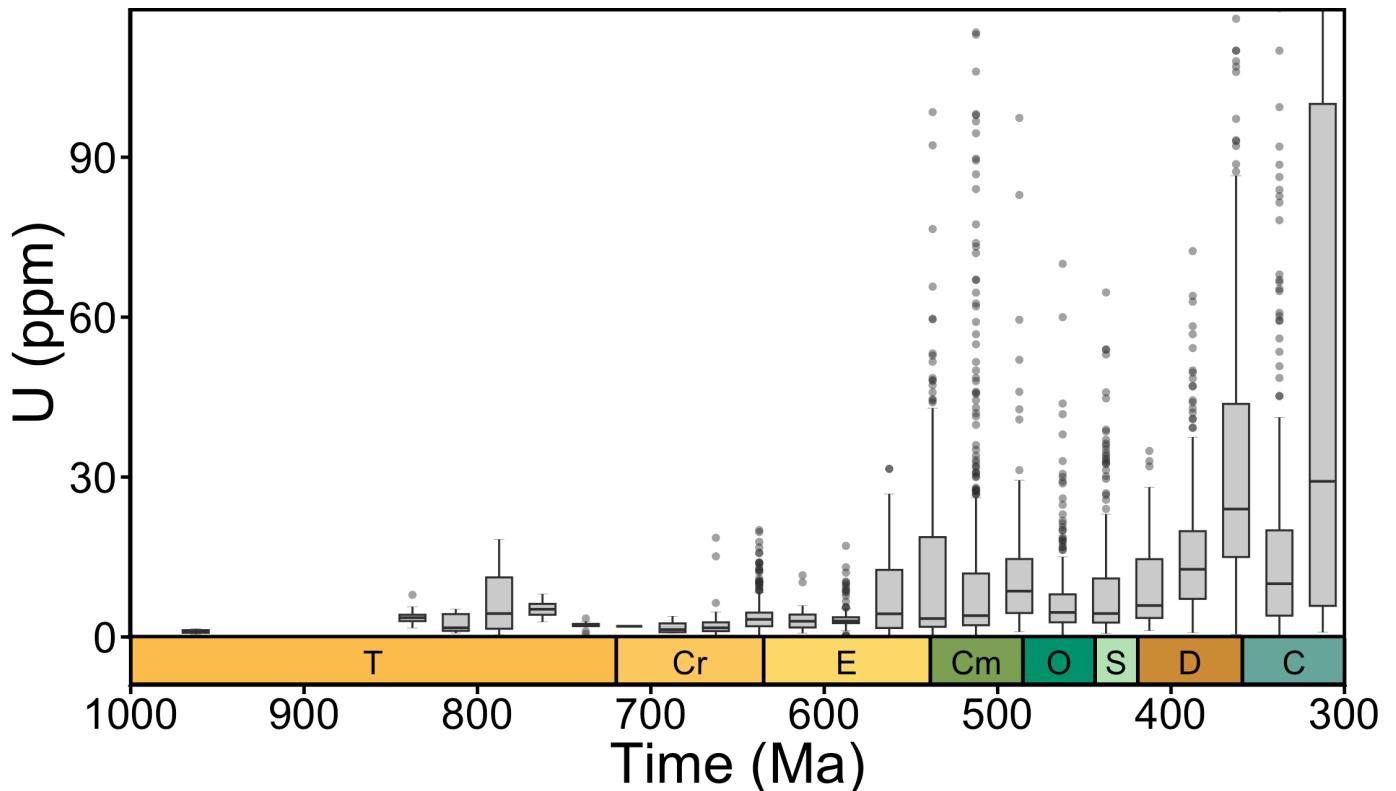
```

U.box.raw.standalone <- ggplot(U.anox.box, aes(x=time.bin, y=U..ppm., group=time.b
in))+
  stat_boxplot(geom = 'errorbar', size=0.6, width = 6, color="grey20", lwd=.1) +
  geom_boxplot(outlier.alpha = .5, outlier.shape=16, outlier.size=2, outlier.color
="grey20", width=15, color="grey20", lwd=0.6, fill="grey80", fatten = 1.3)+
  coord_geo(xlim=c(start_age+1,end_age-1), ylim=c(-.25,118),expand=FALSE,
             pos = as.list(rep("bottom", 1)),
             abbrv=list( T),
             dat = list(periods.edit),
             height = list(unit(2, "lines")),
             size=list(7),
             bord=list(c("left", "bottom", "right")), lwd=as.list(1))+

  theme_minimal()+
  scale_x_reverse(breaks=c(300,400,500,600,700,800,900,1000))+
  #breaks=c(360,400,440,480))+

  ylab("U (ppm)")+
  xlab("Time (Ma)")+
  theme(plot.margin = ggplot2::margin(0,1,1,1,"cm"),panel.border = element_rect(fi
ll=NA,color="black", size=2,linetype="solid"),
        axis.ticks = element_line(size=1),
        axis.line = element_line(lineend = 'square'),
        axis.title = element_text(size=34),
        axis.text = element_text( size=26, color="black"),
        legend.title = element_text(size=24),
        legend.text = element_text( size=18),
        axis.ticks.length = unit(5, "points"),
        legend.position="none",
        panel.grid.major = element_blank(),panel.grid.minor = element_blank())
U.box.raw.standalone

```



Proportion euxinic. Because of the binary nature of our underlying data for proportion euxinic, we overlay a point representing the mean of the data.

```
Fepy.anox.box.sum <- Fepy.anox.box %>%
  group_by(time.bin) %>%
  summarize(euxinic.Fe.median = median(euxinic.Fe, na.rm = TRUE), euxinic.Fe.mean =
mean(euxinic.Fe, na.rm = TRUE))
```

```
Fepy.box.raw.standalone <- ggplot(Fepy.anox.box, aes(x=time.bin, y=euxinic.Fe, gro
up=time.bin))+

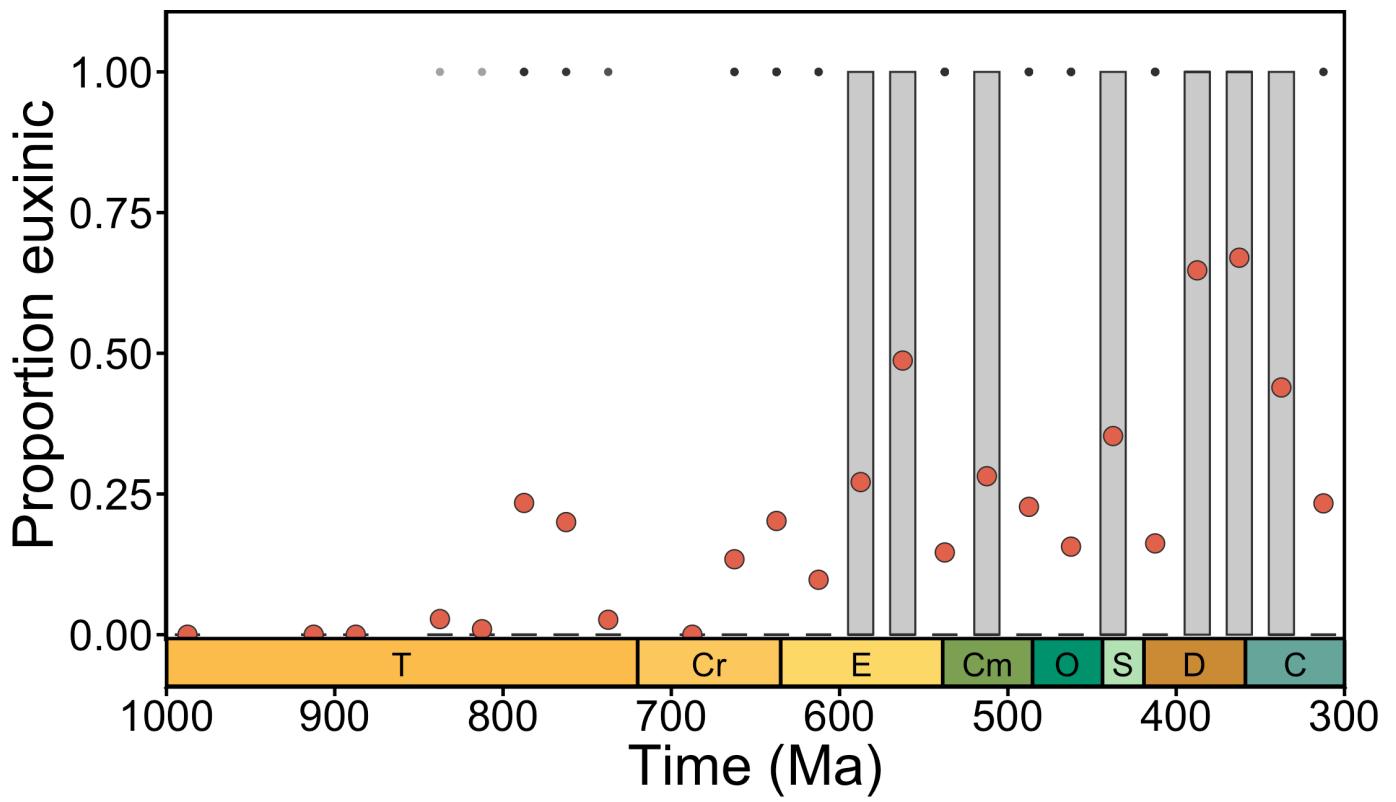
  stat_boxplot(geom ='errorbar', size=0.6, width = 6, color="grey20", lwd=.1) +
  geom_boxplot(outlier.alpha = .5, outlier.shape=16, outlier.size=2, outlier.color
="grey20", width=15, color="grey20", lwd=0.6, fill="grey80", fatten = 1.3)+

  annotate(geom="point", y = Fepy.anox.box.sum$euxinic.Fe.mean, x = Fepy.anox.box.
sum$time.bin, shape = 21, size = 5, color = "grey20", fill = rgb(227, 99, 80, maxC
olorValue = 255), stroke=0.6)+

  coord_geo(xlim=c(start_age+1,end_age-1), ylim=c(-.01,1.11),expand=FALSE,
    pos = as.list(rep("bottom", 1)),
    abbrv=list( T),
    dat = list(periods.edit),
    height = list(unit(2, "lines")),
    size=list(7),
    bord=list(c("left", "bottom", "right")), lwd=as.list(1))+

  theme_minimal()+scale_x_reverse(breaks=c(300,400,500,600,700,800,900,1000))+#bre
aks=c(360,400,440,480))+

  ylab("Proportion euxinic")+xlab("Time (Ma)")+
  theme(plot.margin = ggplot2::margin(0,1,1,1,"cm"),panel.border = element_rect(fi
ll=NA,color="black", size=2,linetype="solid"),
  axis.ticks = element_line(size=1),
  axis.line = element_line(lineend = 'square'),
  axis.title = element_text(size=34),
  axis.text = element_text( size=26, color="black"),
  legend.title = element_text(size=24),
  legend.text = element_text( size=18),
  axis.ticks.length = unit(5, "points"),
  legend.position="none",
  panel.grid.major = element_blank(),panel.grid.minor = element_blank())
Fepy.box.raw.standalone
```

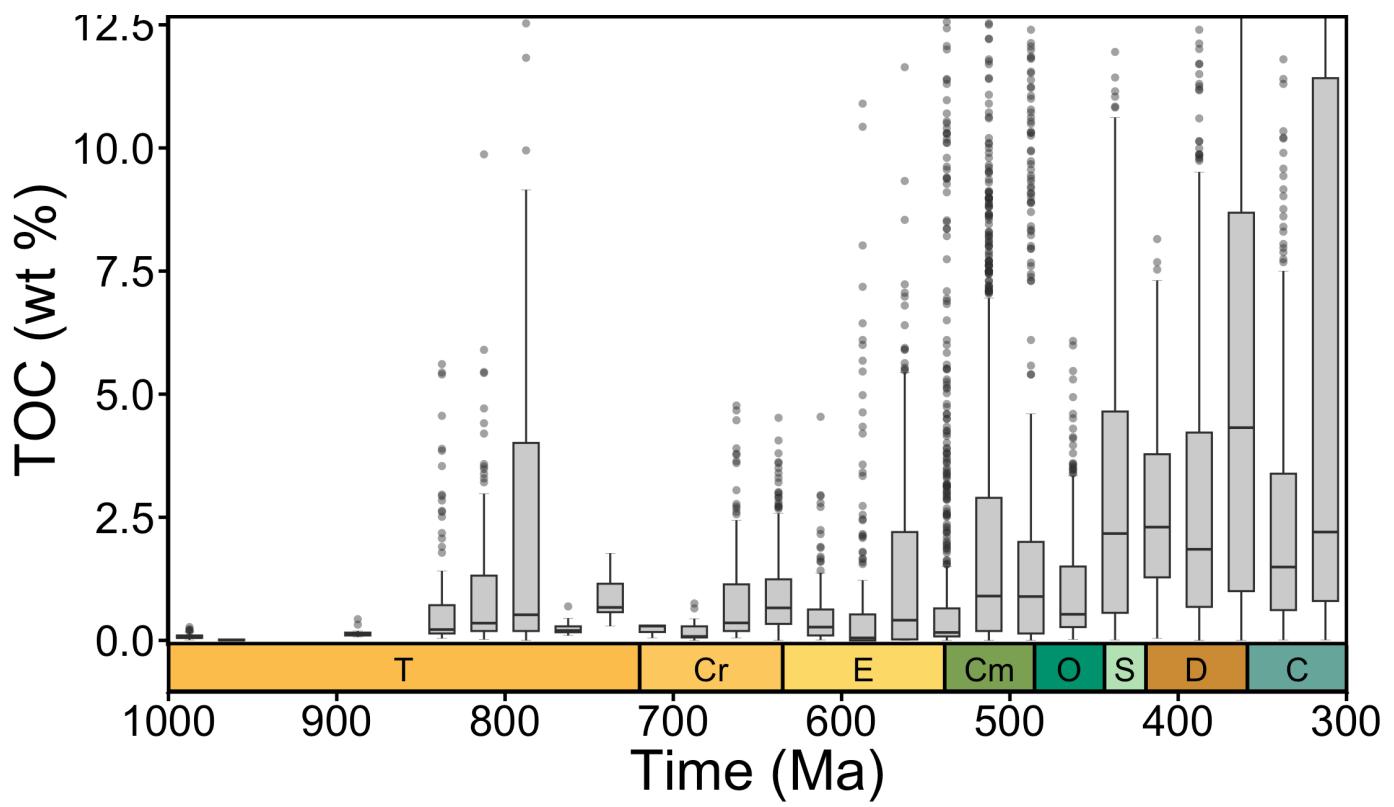


TOC

```

TOC.box.raw.standalone <- ggplot(TOC.all.box, aes(x=time.bin, y=TOC..wt..., group=time.bin))+
  stat_boxplot(geom = 'errorbar', size=0.6, width = 6, color="grey20", lwd=.1) +
  geom_boxplot(outlier.alpha = .5, outlier.shape=16, outlier.size=2, outlier.color = "grey20", width=15, color="grey20", lwd=0.6, fill="grey80", fatten = 1.3) +
  coord_geo(xlim=c(start_age+1,end_age-1), ylim=c(-.1,12.7),expand=FALSE,
            pos = as.list(rep("bottom", 1)),
            abbrv=list( T),
            dat = list(periods.edit),
            height = list(unit(2, "lines")),
            size=list(7),
            bord=list(c("left", "bottom", "right")), lwd=as.list(1)) +
  theme_minimal() + scale_x_reverse(breaks=c(300,400,500,600,700,800,900,1000)) + #breaks=c(360,400,440,480)) +
  ylab("TOC (wt %)") + xlab("Time (Ma)") +
  theme(plot.margin = ggplot2::margin(0,1,1,1,"cm"), panel.border = element_rect(fill=NA,color="black", size=2,linetype="solid"),
        axis.ticks = element_line(size=1),
        axis.line = element_line(lineend = 'square'),
        axis.title = element_text(size=34),
        axis.text = element_text( size=26, color="black"),
        legend.title = element_text(size=24),
        legend.text = element_text( size=18),
        axis.ticks.length = unit(5, "points"),
        legend.position="none",
        panel.grid.major = element_blank(), panel.grid.minor = element_blank())
TOC.box.raw.standalone

```



Generate histograms of the number of samples in each time bin for each analysis

Generate histogram for Mo samples (with axis labels first to confirm all is correct before removing labels).

Note that in two histograms of slightly different widths are plotted for aesthetic purposes, which is what prompts the warning during the `ggplot()` call. This does not impact the heights or centers of the bars in any way.

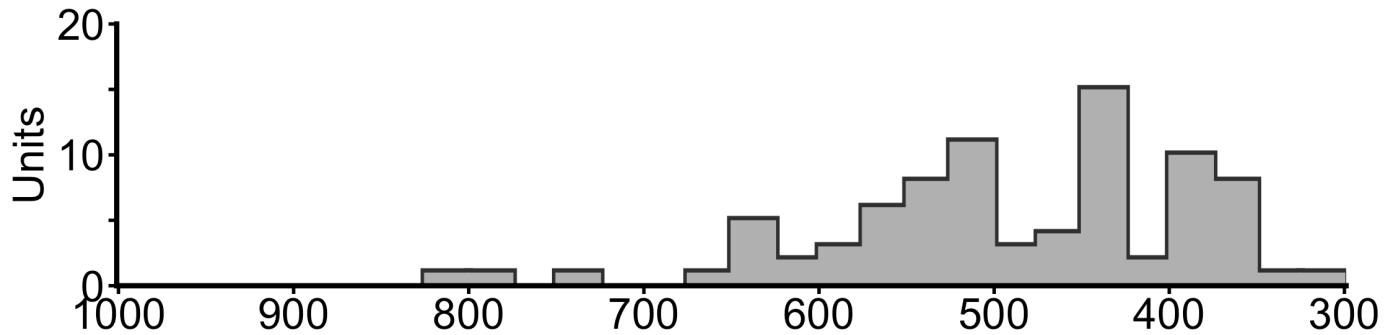
```

Mo.eux.units <- Mo.eux.box %>%
  group_by(long.stratigraphy.name, time.bin) %>%
  tally() %>%
  group_by(time.bin) %>%
  as.data.frame()

Mo.eux.hist.labelled <- ggplot(Mo.eux.units, aes(time.bin))+
  geom_bar(fill="grey70", color="grey20", size=2.5, width=25)+
  geom_bar(fill="grey70", color="grey70", size=0, width=26)+
  coord_cartesian(xlim=c(start_age+1,end_age-1), ylim=c(0,20), expand=FALSE)+
  theme_minimal()+scale_x_reverse(breaks=c(300,400,500,600,700,800,900,1000))+#breaks=c(360,400,440,480))+#breaks=c(300,400,500,600,700,800,900,1000))
  scale_y_continuous(breaks=c(0,5,10,15,20), labels=c("0","",""10","",""20"))+
  ylab("Units")+
  xlab("Time (Ma)")+
  theme(plot.margin = ggplot2::margin(1,1,-1,1,"cm"),# edited margin to show full histogram
        panel.border = element_rect(fill=NA,color=NA,size=2,linetype="solid"),
        #axis.line.x = element_blank(),
        axis.line.y = element_line(lineend = 'square', color="black", size=1.5),
        axis.line.x = element_line(lineend = 'square', color="black", size=1.5), #added in for full labelling
        axis.ticks = element_line(size=1),
        #axis.ticks.x = element_blank(),
        axis.title = element_text(size=26),
        axis.text = element_text( size=26, color="black"),
        legend.text = element_text( size=16),
        legend.title = element_text( size=16),
        legend.position="none",
        legend.background = element_rect(color=NA, fill=NA),
        #axis.title.x = element_blank(),
        #axis.text.x = element_blank(),
        axis.ticks.length = unit(5, "points"),
        panel.grid.major = element_blank(),panel.grid.minor = element_blank())
Mo.eux.hist.labelled

```

Warning: `position_stack()` requires non-overlapping x intervals



Histogram for Mo samples with no labels for plotting.

```

Mo.eux.hist <- ggplot(Mo.eux.units, aes(time.bin))+  

  geom_bar(fill="grey70", color="grey20", size=2.5, width=25)+  

  geom_bar(fill="grey70", color="grey70", size=0, width=26)+  

  coord_cartesian(xlim=c(start_age+1,end_age-1), ylim=c(0,20), expand=FALSE)+  

  theme_minimal()+scale_x_reverse(breaks=c(300,400,500,600,700,800,900,1000))+#bre  

aks=c(360,400,440,480))+  

  scale_y_continuous(breaks=c(0,5,10,15,20), labels=c("0","",""10,"",""20"))+  

  ylab("Units") +  

  xlab("Time (Ma)") +  

  theme(plot.margin = ggplot2::margin(1,1,-0.2,1,"cm"), panel.border = element_rect  

(fill=NA,color=NA, size=2,linetype="solid"),  

  axis.line.x = element_blank(),  

  axis.line.y = element_line(lineend = 'square', color="black", size=1.5),  

  axis.ticks = element_line(size=1),  

  axis.ticks.x = element_blank(),  

  axis.title = element_text(size=26),  

  axis.text = element_text( size=26, color="black"),  

  legend.text = element_text( size=16),  

  legend.title = element_text( size=16),  

  legend.position="none",  

  legend.background = element_rect(color=NA, fill=NA),  

  axis.title.x = element_blank(),  

  axis.text.x = element_blank(),  

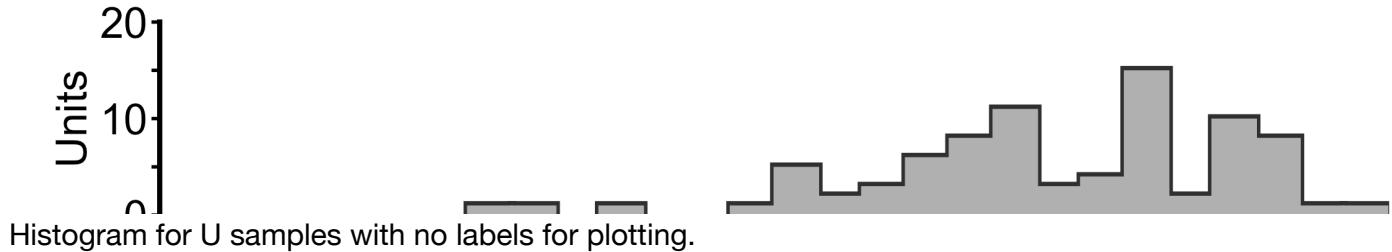
  axis.ticks.length = unit(5, "points"),  

  panel.grid.major = element_blank(), panel.grid.minor = element_blank())  

Mo.eux.hist

```

```
## Warning: `position_stack()` requires non-overlapping x intervals
```



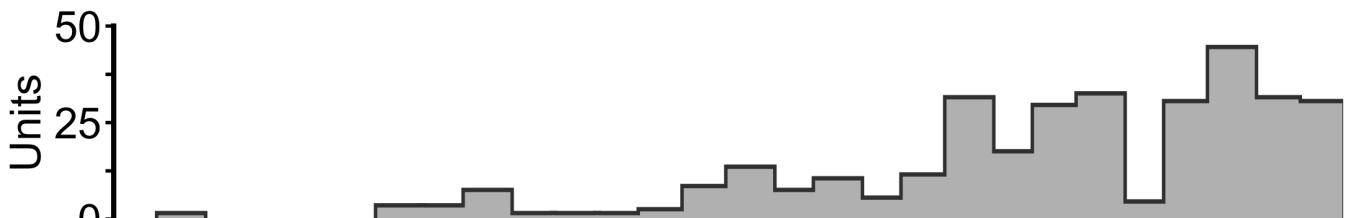
```

U.anox.units <- U.anox.box %>%
  group_by(long.stratigraphy.name, time.bin) %>%
  tally() %>%
  group_by(time.bin) %>%
  as.data.frame()

U.anox.hist <- ggplot(U.anox.units, aes(time.bin))+
  geom_bar(fill="grey70", color="grey20", size=2.5, width=25)+
  geom_bar(fill="grey70", color="grey70", size=0, width=26)+
  coord_cartesian(xlim=c(start_age+1,end_age-1), ylim=c(0,50), expand=FALSE)+
  theme_minimal()+scale_x_reverse(breaks=c(300,400,500,600,700,800,900,1000))+#breaks=c(360,400,440,480))+
  scale_y_continuous(breaks=c(0,12.5,25,37.5,50), labels=c("0","",""25,"","","50"))
+
  ylab("Units")+
  xlab("Time (Ma)")+
  theme(plot.margin = ggplot2::margin(1,1,-0.2,1,"cm"),panel.border = element_rect(
    fill=NA,color=NA, size=2,linetype="solid"),
    axis.line.x = element_blank(),
    axis.line.y = element_line(lineend = 'square', color="black", size=1.5),
    axis.ticks = element_line(size=1),
    axis.ticks.x = element_blank(),
    axis.title = element_text(size=26),
    axis.text = element_text( size=26, color="black"),
    legend.text = element_text( size=16),
    legend.title = element_text( size=16),
    legend.position="none",
    legend.background = element_rect(color=NA, fill=NA),
    axis.title.x = element_blank(),
    axis.text.x = element_blank(),
    axis.ticks.length = unit(5, "points"),
    panel.grid.major = element_blank(),panel.grid.minor = element_blank())
U.anox.hist

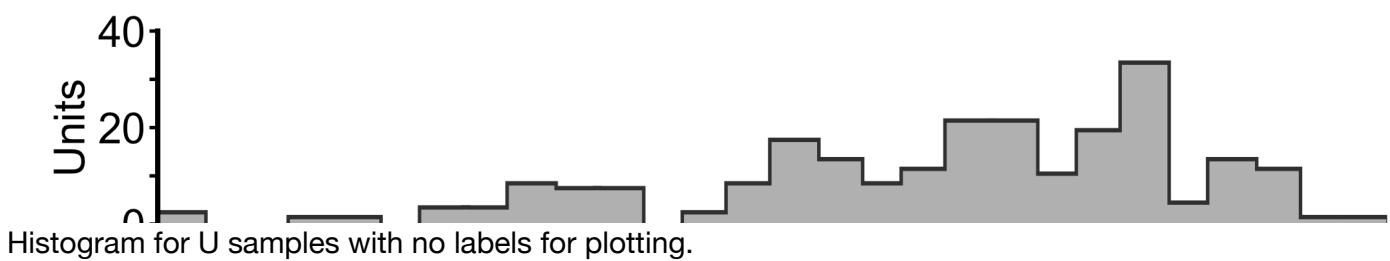
```

```
## Warning: `position_stack()` requires non-overlapping x intervals
```

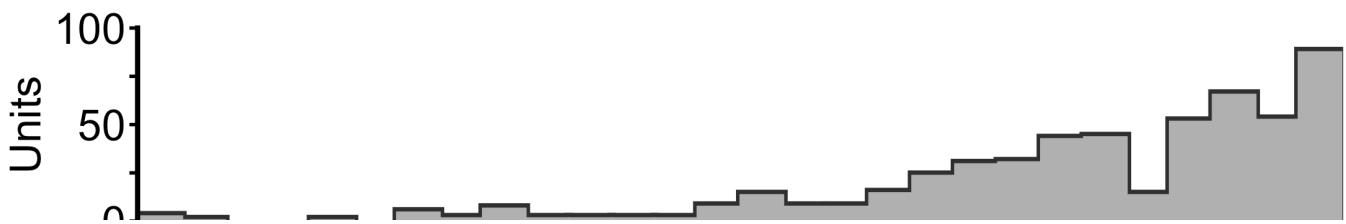


Histogram for proportion euxinic samples with no labels for plotting.

Warning: `position_stack()` requires non-overlapping x intervals



Warning: `position_stack()` requires non-overlapping x intervals



Plot summary figure of raw data

Define box and whisker plot panels for composite.

```

Mo.box.raw.composite <- ggplot(Mo.eux.box, aes(x=time.bin, y=Mo..ppm., group=time.
bin))+  

  stat_boxplot(geom ='errorbar', size=0.6, width = 6, color="grey20", lwd=.1) +  

  geom_boxplot(outlier.alpha = .5, outlier.shape=16, outlier.size=2, outlier.color  

="grey20", width=15, color="grey20", lwd=0.6, fill="grey80", fatten = 1.3)+  

  coord cartesian(xlim=c(start age1.end age-1), ylim=c(-.5,325),expand=FALSE)+
```

```

theme_minimal() + scale_x_reverse(breaks=c(300, 400, 500, 600, 700, 800, 900, 1000)) + #breaks=c(360, 400, 440, 480)) +
  ylab("Mo (ppm)") + xlab("Time (Ma)") +
  theme(plot.margin = ggplot2::margin(0, 1, 0, 1, "cm"), panel.border = element_rect(fill=NA, color="black", size=2, linetype="solid"),
        axis.ticks = element_line(size=1),
        axis.line = element_line(lineend = 'square'),
        axis.title = element_text(size=34),
        axis.text = element_text( size=26, color="black"),
        legend.title = element_text(size=24),
        legend.text = element_text( size=18),
        axis.title.x = element_blank(),
        axis.text.x = element_blank(),
        axis.ticks.length = unit(5, "points"),
        legend.position="none",
        panel.grid.major = element_blank(), panel.grid.minor = element_blank())

```

```

U.box.raw.composite <- ggplot(U.anox.box, aes(x=time.bin, y=U..ppm., group=time.bin)) +
  stat_boxplot(geom = 'errorbar', size=0.6, width = 6, color="grey20", lwd=.1) +
  geom_boxplot(outlier.alpha = .5, outlier.shape=16, outlier.size=2, outlier.color = "grey20", width=15, color="grey20", lwd=0.6, fill="grey80", fatten = 1.3) +
  coord_geo(xlim=c(start_age+1, end_age-1), ylim=c(-.25, 188), expand=FALSE,
             pos = as.list(rep("bottom", 1)),
             abbrv=list( T),
             dat = list(periods.edit),
             height = list(unit(2, "lines")),
             size=list(7),
             bord=list(c("left", "bottom", "right")), lwd=as.list(1)) +
  theme_minimal() + scale_x_reverse(breaks=c(300, 400, 500, 600, 700, 800, 900, 1000)) + #breaks=c(360, 400, 440, 480)) +
  ylab("U (ppm)") + xlab("Time (Ma)") +
  theme(plot.margin = ggplot2::margin(0, 1, 0, 1, "cm"), panel.border = element_rect(fill=NA, color="black", size=2, linetype="solid"),
        axis.ticks = element_line(size=1),
        axis.line = element_line(lineend = 'square'),
        axis.title = element_text(size=34),
        axis.text = element_text( size=26, color="black"),
        legend.title = element_text(size=24),
        legend.text = element_text( size=18),
        axis.ticks.length = unit(5, "points"),
        legend.position="none",
        panel.grid.major = element_blank(), panel.grid.minor = element_blank())

```

```

Fepy.box.raw.composite <- ggplot(Fepy.anox.box, aes(x=time.bin, y=euxinic.Fe, group=time.bin)) +
  stat_boxplot(geom = 'errorbar', size=0.6, width = 6, color="grey20", lwd=.1) +
  geom_boxplot(outlier.alpha = .5, outlier.shape=16, outlier.size=2, outlier.color = "grey20", width=15, color="grey20", lwd=0.6, fill="grey80", fatten = 1.3) +
  annotate(geom="point", y = Fepy.anox.box.sum$euxinic.Fe.mean, x = Fepy.anox.box.sum$time.bin, shape = 21, size = 5, color = "grey20", fill = rgb(227, 99, 80, maxColorValue = 255), stroke=0.6) +
  coord_cartesian(c(start_age+1, end_age-1), ylim=c(-.01, 1.11), expand=FALSE) +

```

```

theme_minimal() + scale_x_reverse(breaks=c(300, 400, 500, 600, 700, 800, 900, 1000)) + #breaks=c(360, 400, 440, 480)) +
  ylab("Proportion euxinic") + xlab("Time (Ma)") +
  theme(plot.margin = ggplot2::margin(0, 1, 0, 1, "cm"), panel.border = element_rect(fill=NA, color="black", size=2, linetype="solid"),
        axis.ticks = element_line(size=1),
        axis.line = element_line(lineend = 'square'),
        axis.title = element_text(size=34),
        axis.text = element_text( size=26, color="black"),
        legend.title = element_text(size=24),
        legend.text = element_text( size=18),
        axis.title.x = element_blank(),
        axis.text.x = element_blank(),
        axis.ticks.length = unit(5, "points"),
        legend.position="none",
        panel.grid.major = element_blank(), panel.grid.minor = element_blank())

```



```

TOC.box.raw.composite <- ggplot(TOC.all.box, aes(x=time.bin, y=TOC..wt.., group=ti
me.bin)) +
  stat_boxplot(geom = 'errorbar', size=0.6, width = 6, color="grey20", lwd=.1) +
  geom_boxplot(outlier.alpha = .5, outlier.shape=16, outlier.size=2, outlier.color
="grey20", width=15, color="grey20", lwd=0.6, fill="grey80", fatten = 1.3) +
  coord_geo(xlim=c(start_age+1, end_age-1), ylim=c(-.1, 14.7), expand=FALSE,
             pos = as.list(rep("bottom", 1)),
             abbrv=list( T),
             dat = list(periods.edit),
             height = list(unit(2, "lines")),
             size=list(7),
             bord=list(c("left", "bottom", "right")), lwd=as.list(1)) +
  theme_minimal() + scale_x_reverse(breaks=c(300, 400, 500, 600, 700, 800, 900, 1000)) +
  ylab("TOC (wt %)") + xlab("Time (Ma)") +
  theme(plot.margin = ggplot2::margin(0, 1, 1, 1, "cm"), panel.border = element_rect(fill=NA, color="black", size=2, linetype="solid"),
        axis.ticks = element_line(size=1),
        axis.line = element_line(lineend = 'square'),
        axis.title = element_text(size=34),
        axis.text = element_text( size=26, color="black"),
        legend.title = element_text(size=24),
        legend.text = element_text( size=18),
        axis.ticks.length = unit(5, "points"),
        legend.position="none",
        panel.grid.major = element_blank(), panel.grid.minor = element_blank())

```

Combine 4 box and whisker plot panels and associated histograms.

```

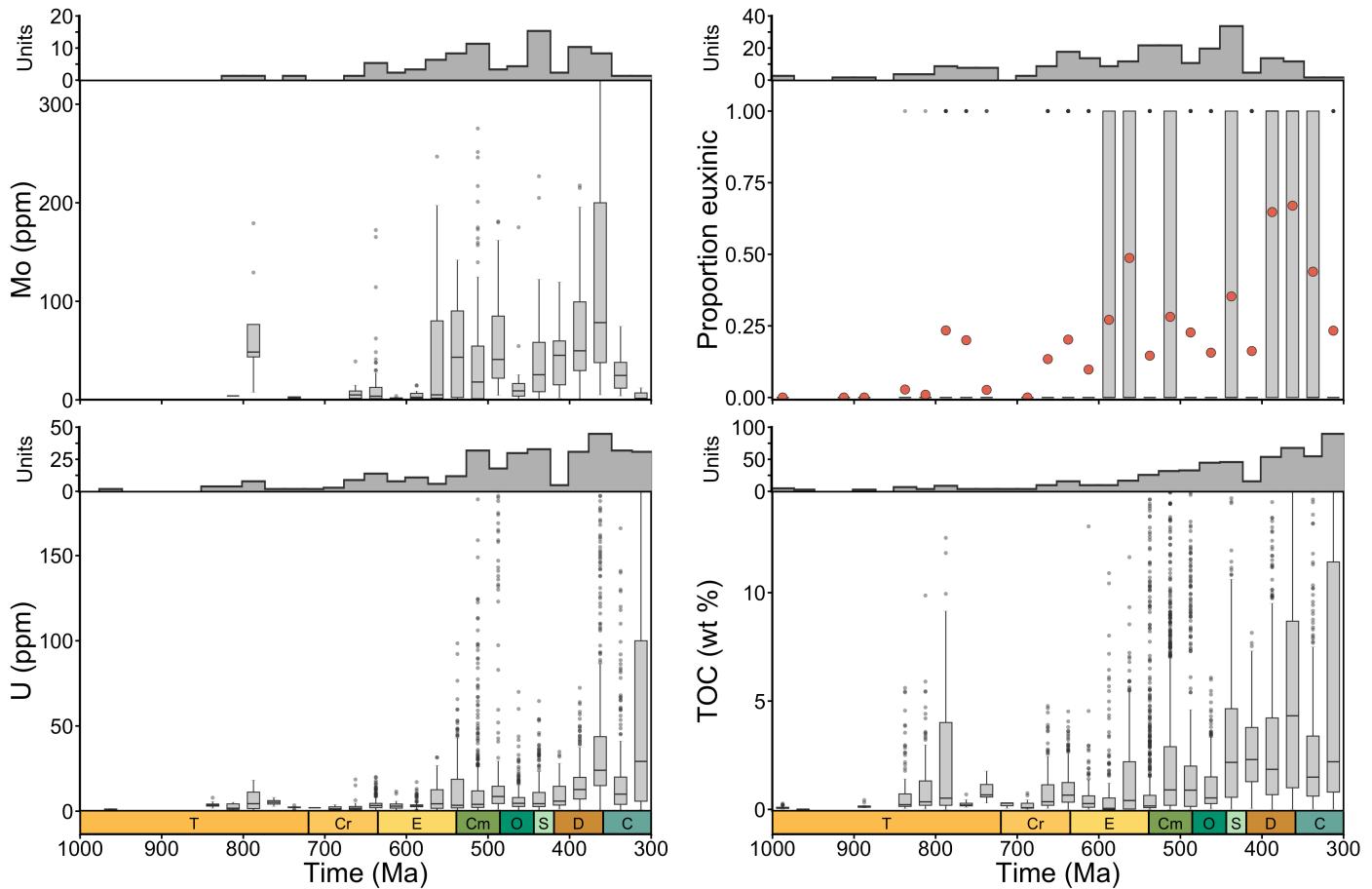
box.fig.composite <- ggarrange2(Mo.eux.hist,
                                Fepy.anox.hist,
                                Mo.box.raw.composite,
                                Fepy.box.raw.composite,
                                U.anox.hist,
                                TOC.all.hist,
                                U.box.raw.composite,
                                TOC.box.raw.composite,
                                ncol = 2,
                                heights = c(0.2,1,0.2,1)
)

```

```

## Warning: `position_stack()` requires non-overlapping x intervals
## `position_stack()` requires non-overlapping x intervals
## `position_stack()` requires non-overlapping x intervals
## `position_stack()` requires non-overlapping x intervals

```



```

ggsave(file="Figure Sx Raw data box and whisker plots 20240208.pdf", box.fig.composite, height=16, width=24)

```

4. Spatial-temporal bootstrap

Before beginning any bootstrap we set seed for random number generation to ensure reproducibility.

```
set.seed(1993)
```

Define bootstrap function to bin each dataset

```
spatial.scale <- 111*0.5 # convert 0.5 degrees (as in Mehra et al.) to approximate value in km, based on degree to km conversion at equator. (https://solarsystem.nasa.gov/basics/chapter2-1/#:~:text=One%20degree%20of%20latitude%20equals,definition%20exactly%2060%20nautical%20miles.)
```

```
age.scale <- 10*(25/4000) # Age scale set to match the age scale used in Mehra et al. but with 25 Myr bins rather than the 4000 Myr timespan used in Mehra et al.
```

```
temp.spat.boot <- function(data, var, reps, bin_size, temp.spat.weight = TRUE, return.weights = FALSE, spatialScale, ageScale){  
  # data - a subset dataframe of the same format generated for the bootstrap analyses in this study.  
  # var - must be column name of proxy of interest within quotation marks.  
  # reps - number of repetitions to perform in bootstrap analyses.  
  # bin_size - bin size (same usage as elsewhere in code)  
  # temp.spat.weight - this is set to TRUE by default, so that the function uses the spatiotemporal weighting algorithm in the bootstrapped means. If set to FALSE, the function just generates bootstrapped means from the data, with no spatiotemporal weighting.  
  # return.weights - this is set to FALSE by default (for the version of this function that generates bootstrapped means). If set to TRUE, instead returns original "data" dataframe including the spatial-temporal weights.  
  print(paste("Spatial scale is", spatialScale))  
  print(paste("Age scale is", ageScale))  
  
  # initiate vector to store the distribution of bootstrapped means in, and initiate a vector (which will be displaced by a dataframe) to store the new version of 'data' in if return.weights is TRUE.  
  bootmeans <- as.numeric()  
  new.data <- as.numeric()  
  # Loop through time bins in sequence (youngest to oldest) and generate bootstrapped mean values based on spatial-temporal weighting algorithm  
  for (bin in seq(min(data$time.bin), max(data$time.bin), by = bin_size)){  
  
    # Filter proxy dataset to include only samples from within the time bin targeted in this loop.  
    timebin.data <- data %>%  
      filter(time.bin == bin)  
  
    # If there are no samples in the filtered time bin dataset and you are generating bootstrapped means, store NAs in the summary dataframe for this timestep. If return.weights is turned on, add a row of NAs but change time.bin to correct bin (useful for map later).  
    if(nrow(timebin.data) == 0){  
      print(paste("No samples in", bin, "Ma bin."))  
      if(return.weights == FALSE){
```

```

# If generating bootstrapped means, generate NAs
bootmeans <- rbind(bootmeans, cbind(rep(NA, reps), rep(bin, reps)))
}

if(return.weights == TRUE){
# If return.weights == TRUE, add a row of NAs
new.data <- rbind(new.data, rep(NA, ncol(new.data)))
# Then assign the correct timebin.
new.data$time.bin[nrow(new.data)] <- bin
}
}else{
# Else - if all samples in the time bin are of the same interpreted age and
from the same geographic location, then assign all samples equal weight.
if(max(timebin.data$interpreted.age, na.rm = T) - min(timebin.data$interpret
ed.age, na.rm = T) == 0 &
   max(timebin.data$site.longitude, na.rm = T) - min(timebin.data$site.long
itude, na.rm = T) == 0 &
   max(timebin.data$site.latitude, na.rm = T) - min(timebin.data$site.latit
ude, na.rm = T) == 0){

equal.sample.weight <- 1
timebin.data$sample.weight <- equal.sample.weight
timebin.data$sample.P <- equal.sample.weight

}else {
# Else - if none of the above scenarios are true (this should be the case fo
r most time bins for most proxy datasets this
# technique is used on) then begin by initiating a new column of sample weig
hts (assigned NAs initially) and then loop through samples
# assigning them weights based on the spatial-temporal weighting method of M
ehra et al. 2021.

timebin.data$sample.weight <- NA

# Loop through rows in time bin dataset.
for(row in 1:nrow(timebin.data)) {

## REPLICATION OF MEHRA CODE BEGINS HERE - CHECK ALL!

# CHECK - should lp always be 2? Check this.
lp <- 2

# Weighting algorithm from Mehra et al.

# load geosphere package to compute great circle distances.
library(geosphere)

# in package geosphere there are many ways to compute great circle distances
<- distGeo() is supposed to be most accurate (could revise)
# p1 - the coordinates of the sample in the current row in the loop.
# p2 - the coordinates of all other samples in the time bin
p1 <- as.matrix(cbind(timebin.data$site.longitude[row], timebin.data$site.la
titude[row]))
p2 <- as.matrix(cbind(timebin.data$site.longitude[-row], timebin.data$site.la
titude[-row]))
```

```

    # Calculate great circle distances between the sample in current row (in loop)
    # and all other samples in time bin
    spaceDistances.km <- spDistsN1(p2, p1, longlat = TRUE)

    # Calculate age differences between the sample in current row (in loop) and
    # all other samples in time bin
    ageDistances <- abs(timebin.data$interpreted.age[row] - timebin.data$interpreted.age[-row])

    # Calculate distance weighting
    # CHECK - against Mehra eqns.
    distanceWeighting <- 1/((spaceDistances.km / spatialScale)^lp + 1.0)

    # Calculate age weighting
    # CHECK - against Mehra eqns.
    ageWeighting <- 1/((ageDistances/ageScale)^lp + 1.0)

    # Calculate combined weighting
    # CHECK - against Mehra eqns.
    sample.weight <- sum(distanceWeighting + ageWeighting, na.rm = T)

    # Store sample weight in timebin dataframe
    timebin.data$sample.weight[row] <- sample.weight

}

# Calculate probability with which sample should be drawn.
timebin.data$sample.P <- NA
for(row in 1:nrow(timebin.data)){
  timebin.data$sample.P[row] <- 1/((timebin.data$sample.weight[row] * median(0.2
  /timebin.data$sample.weight, na.rm=T))+1)
}
}

if(return.weights == FALSE){

  # Bootstrap mean values of proxy of interest based upon weightings calculated
  # above.

  # Note that "var" is vectorized before the bootstrap because it is pointed to
  # as an individual column in the function call.
  var.vec <- timebin.data[, var]

  # if temp.spat.weight = TRUE - then use weights to generate bootstrapped mean
  if(temp.spat.weight == TRUE){
    print("Using temporal-spatial weighting algorithm")
    timebin.bootmeans <- one.boot(data = var.vec, FUN = mean, R = reps, weights =
    timebin.data$sample.P)
  }
  # if temp.spat.weight = FALSE - then DO NOT use weights to generate bootstrapped mean
  if(temp.spat.weight == FALSE){
    print("NOT using temporal-spatial weighting algorithm")
    timebin.bootmeans <- one.boot(data = var.vec, FUN = mean, R = reps)
  }
}

```

```

}

# Combine bootstrapped means with other time bins.
bootmeans <- rbind(bootmeans, cbind(timebin.bootmeans$t, rep(bin, nrow(timebin
.bootmeans$t)))))

# If return.weights == TRUE, assemble "data" dataframe with calculated weights
# in a new column.
}else if(return.weights == TRUE){
  new.data <- rbind(new.data, timebin.data)
}

}

}

if(return.weights == FALSE){
  bootmeans <- as.data.frame(bootmeans)
  names(bootmeans) <- c("boot.mean", "time.bin")
  return(bootmeans)
}
if(return.weights == TRUE){
  return(new.data)
}
}

```

Test weighting function

To test the weighting function, we will use the “return.weights” option in the function to return the weights assigned in the function and then plot maps of weights for each time bin. We do this for each of the proxies plotted in the primary results.

Generate weights for molybdenum.

```
Molybdenum.weights <- temp.spat.boot(data = Mo.eux.box, var = "Mo..ppm.", reps = 1
000, bin_size = bin_size, return.weights = TRUE, spatialScale = spatial.scale, age
Scale = age.scale)
```

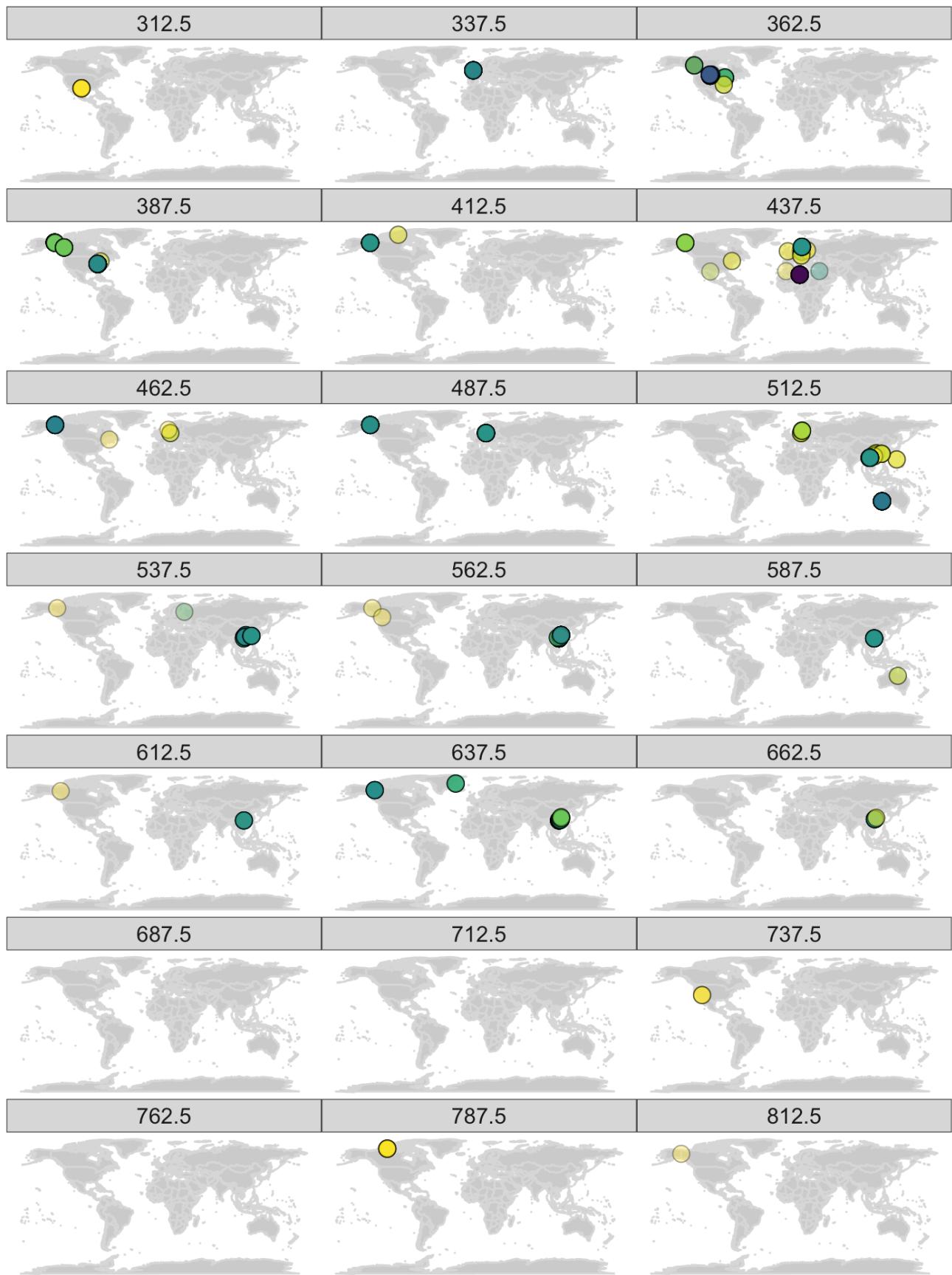
```
## [1] "Spatial scale is 55.5"
## [1] "Age scale is 0.0625"
## [1] "No samples in 687.5 Ma bin."
## [1] "No samples in 712.5 Ma bin."
## [1] "No samples in 762.5 Ma bin."
```

Plot map of molybdenum weights.

```
map <- ggplot() +
  borders("world", colour = "gray85", fill = "gray80") +
  theme_map() +
  geom_point(aes(x = site.longitude, y = site.latitude,
                 fill = sample.P),
             data = Molybdenum.weights,
             alpha = .4,
             shape=21, size=4) +
  facet_wrap(vars(time.bin), ncol=3) +
  scale_fill_viridis_c() +
  labs(fill = 'weight') +
  theme(plot.margin = ggplot2::margin(1,1,1,1,"cm"),
        panel.border = element_rect(fill=NA,color=NA, size=2,linetype="solid"),
        legend.title = element_text(size=20),
        legend.text = element_text(size=16),
        legend.position="top",
        strip.text.x = element_text(size = 12)) +
  guides(fill = guide_legend(override.aes = list(alpha = 1)))
map
```

```
## Warning: Removed 3 rows containing missing values (`geom_point()`).
```

weight ● 0.7 ● 0.8 ● 0.9 ● 1.0



Generate weights for uranium.

```
Uranium.weights <- temp.spat.boot(data = U.anox.box, var = "U..ppm.", reps = 1000,
bin_size = bin_size, return.weights = TRUE, spatialScale = spatial.scale, ageScale
= age.scale)
```

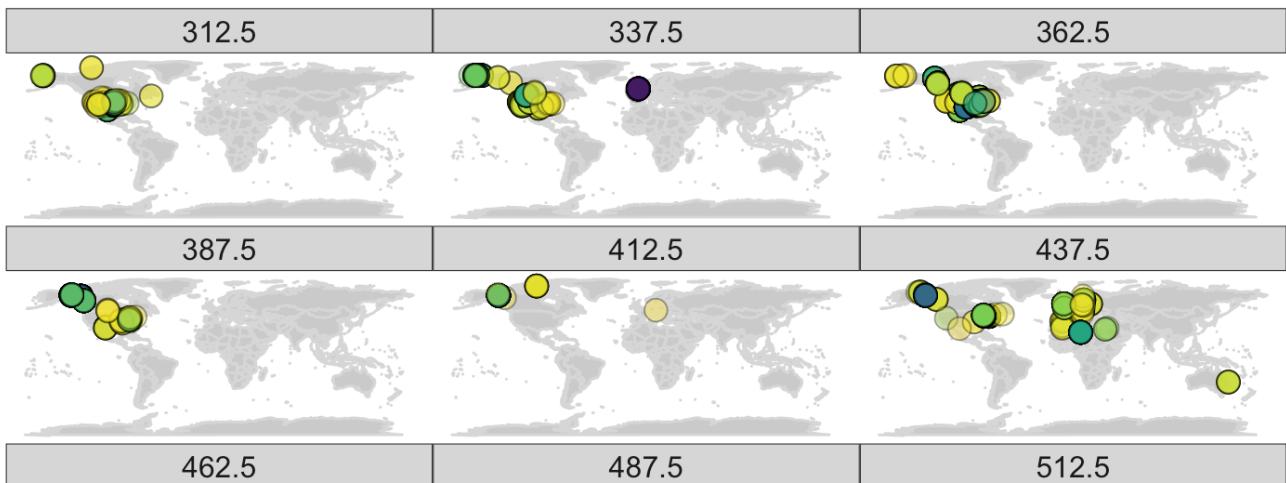
```
## [1] "Spatial scale is 55.5"
## [1] "Age scale is 0.0625"
## [1] "No samples in 862.5 Ma bin."
## [1] "No samples in 887.5 Ma bin."
## [1] "No samples in 912.5 Ma bin."
## [1] "No samples in 937.5 Ma bin."
```

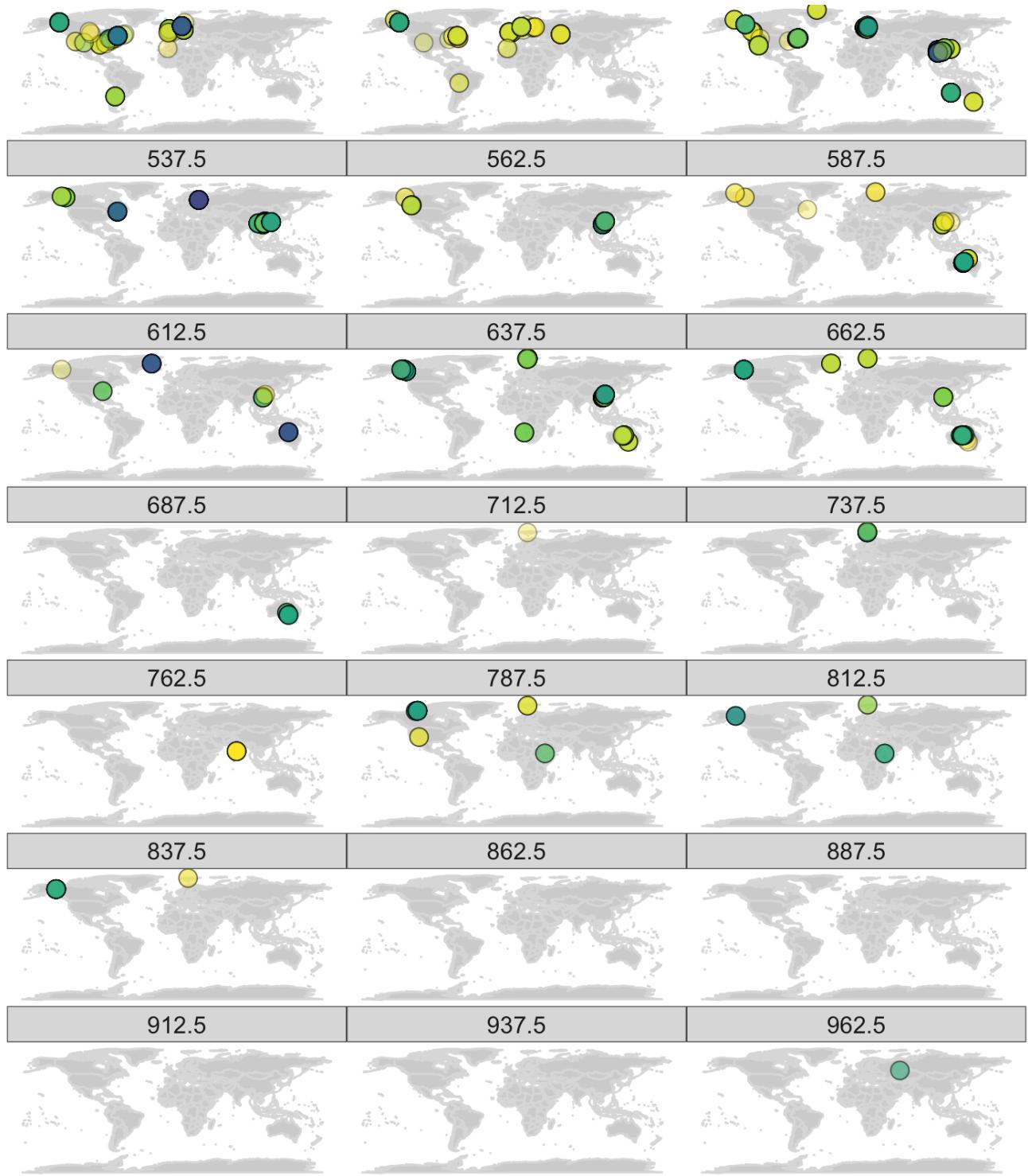
Plot map of uranium weights.

```
map <- ggplot() +
  borders("world", colour = "gray85", fill = "gray80") +
  theme_map() +
  geom_point(aes(x = site.longitude, y = site.latitude,
                 fill = sample.P),
             data = Uranium.weights,
             alpha = .4,
             shape=21, size=4) +
  facet_wrap(vars(time.bin), ncol=3) +
  scale_fill_viridis_c() +
  labs(fill = 'weight') +
  theme(plot.margin = ggplot2::margin(1,1,1,1,"cm"),
        panel.border = element_rect(fill=NA,color=NA, size=2,linetype="solid"),
        legend.title = element_text(size=20),
        legend.text = element_text(size=16),
        legend.position="top",
        strip.text.x = element_text(size = 12)) +
  guides(fill = guide_legend(override.aes = list(alpha = 1)))
map
```

```
## Warning: Removed 4 rows containing missing values (`geom_point()`).
```

weight ● 0.6 ● 0.7 ● 0.8 ● 0.9 ● 1.0





Generate weights for proportion euxinic.

```
Eux.Fe.weights <- temp.spat.boot(data = Fepy.anox.box, var = "euxinic.Fe", reps = 1000, bin_size = bin_size, return.weights = TRUE, spatialScale = spatial.scale, ageScale = age.scale)
```

```

## [1] "Spatial scale is 55.5"
## [1] "Age scale is 0.0625"
## [1] "No samples in 712.5 Ma bin."
## [1] "No samples in 862.5 Ma bin."
## [1] "No samples in 937.5 Ma bin."
## [1] "No samples in 962.5 Ma bin."

```

Plot map of proportion euxinic weights.

```

map <- ggplot() +
  borders("world", colour = "gray85", fill = "gray80") +
  theme_map() +
  geom_point(aes(x = site.longitude, y = site.latitude,
                 fill = sample.P),
             data = Eux.Fe.weights,
             alpha = .4,
             shape=21, size=4) +
  facet_wrap(vars(time.bin), ncol=3) +
  scale_fill_viridis_c() +
  labs(fill = 'weight') +
  theme(plot.margin = ggplot2::margin(1,1,1,1,"cm"),
        panel.border = element_rect(fill=NA,color=NA, size=2,linetype="solid"),
        legend.title = element_text(size=20),
        legend.text = element_text(size=16),
        legend.position="top",
        strip.text.x = element_text(size = 12)) +
  guides(fill = guide_legend(override.aes = list(alpha = 1)))
map

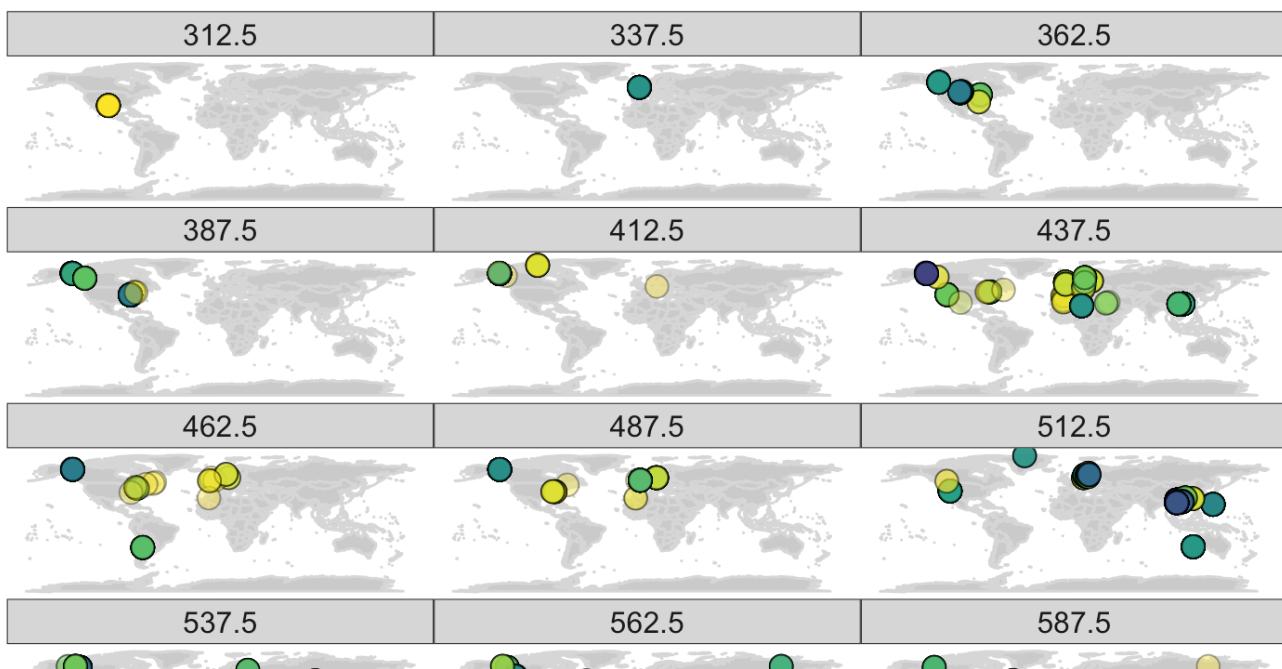
```

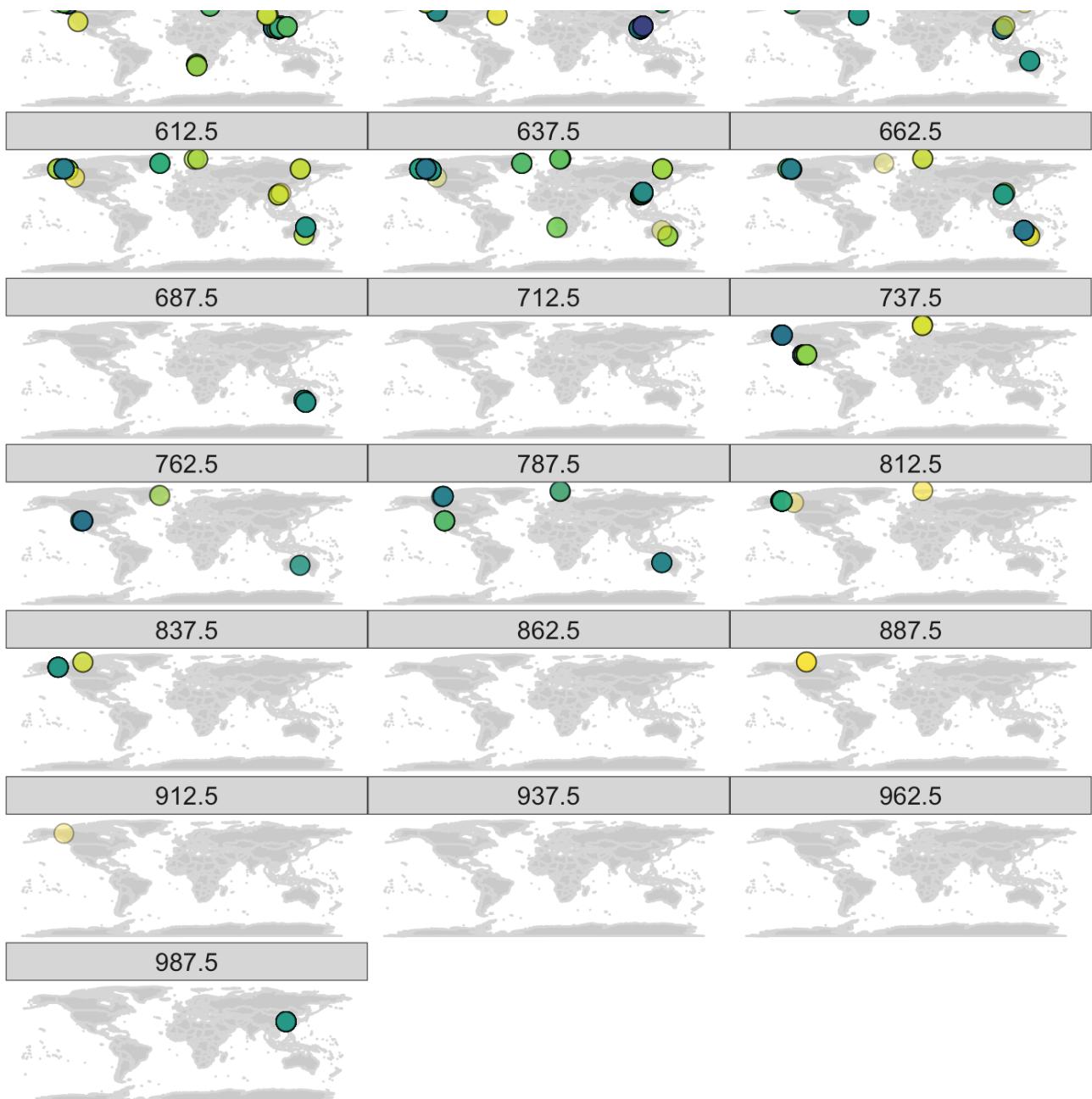
```

## Warning: Removed 4 rows containing missing values (`geom_point()`).

```

weight ● 0.7 ● 0.8 ● 0.9 ● 1.0





Generate weights for TOC

```
TOC.weights <- temp.spat.boot(data = TOC.all.box, var = "TOC..wt..", reps = 1000,
bin_size = bin_size, return.weights = TRUE, spatialScale = spatial.scale, ageScale
= age.scale)
```

```
## [1] "Spatial scale is 55.5"
## [1] "Age scale is 0.0625"
## [1] "No samples in 862.5 Ma bin."
## [1] "No samples in 912.5 Ma bin."
## [1] "No samples in 937.5 Ma bin."
```

Plot map of TOC weights.

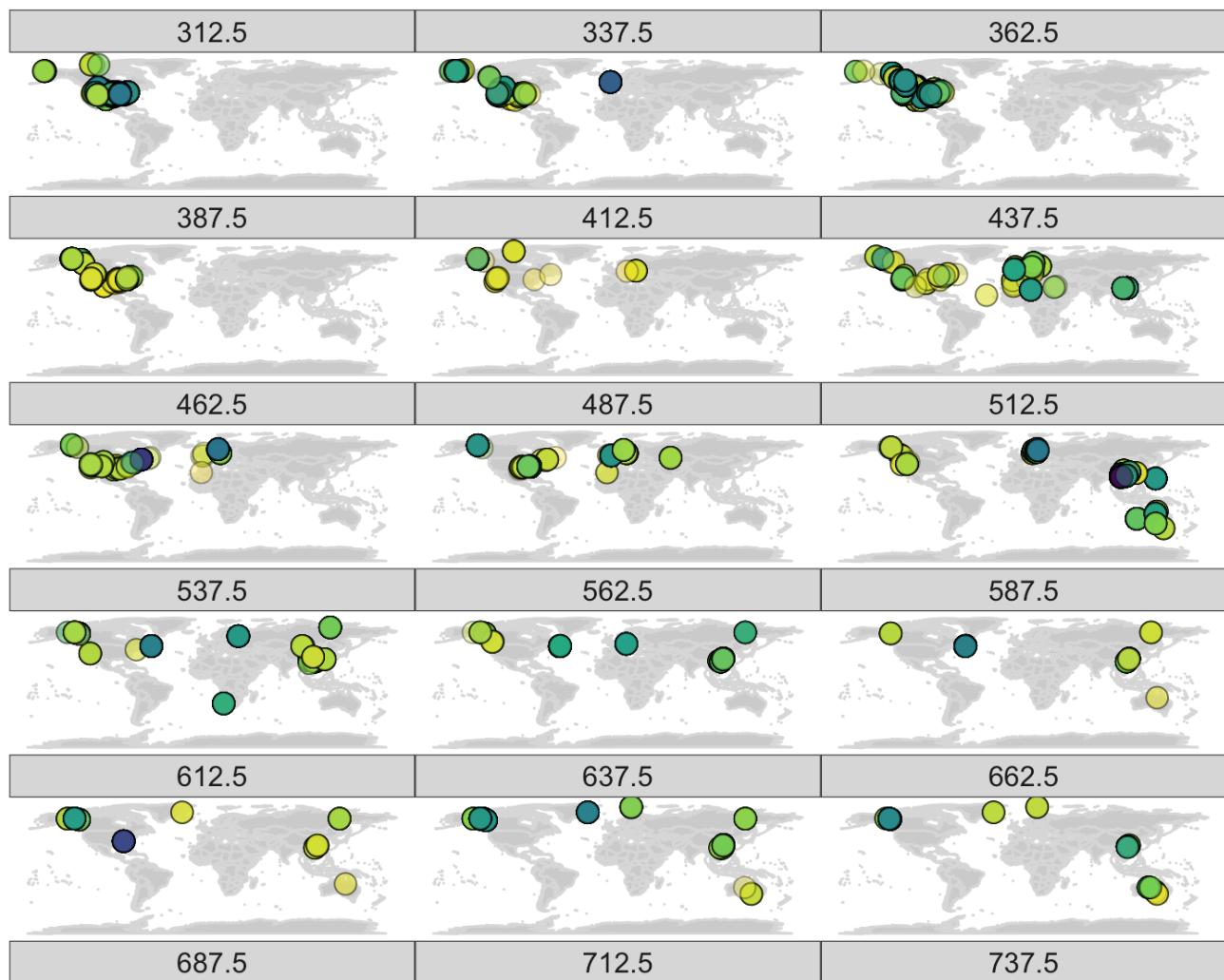
```

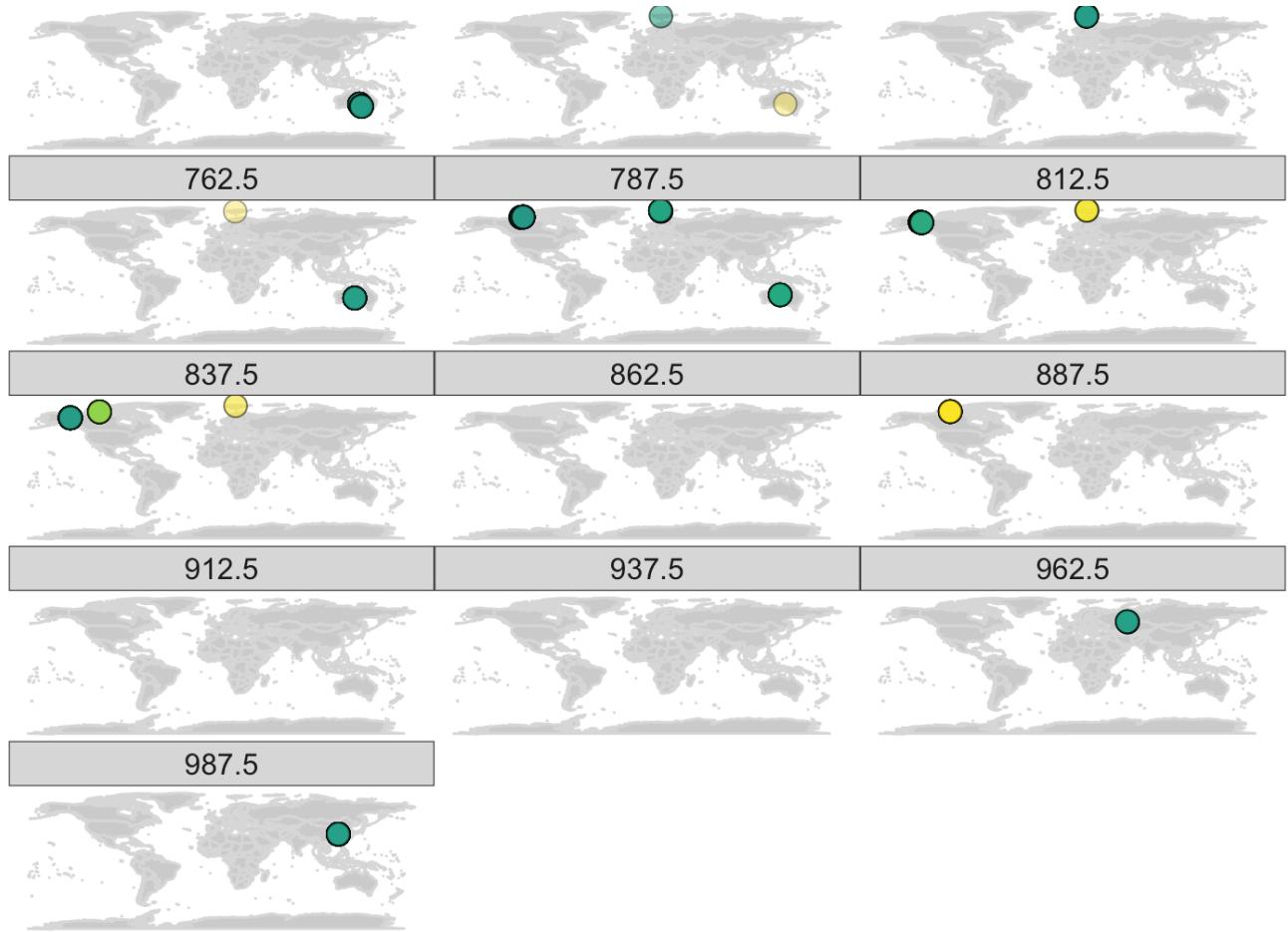
map <- ggplot() +
  borders("world", colour = "gray85", fill = "gray80") +
  theme_map() +
  geom_point(aes(x = site.longitude, y = site.latitude,
                 fill = sample.P),
             data = TOC.weights,
             alpha = .4,
             shape=21, size=4) +
  facet_wrap(vars(time.bin), ncol=3) +
  scale_fill_viridis_c() +
  labs(fill = 'weight') +
  theme(plot.margin = ggplot2::margin(1,1,1,1,"cm"),
        panel.border = element_rect(fill=NA,color=NA, size=2,linetype="solid"),
        legend.title = element_text(size=20),
        legend.text = element_text(size=16),
        legend.position="top",
        strip.text.x = element_text(size = 12)) +
  guides(fill = guide_legend(override.aes = list(alpha = 1)))
map

```

Warning: Removed 3 rows containing missing values (`geom_point()`).

weight ● 0.7 ● 0.8 ● 0.9 ● 1.0





5. Primary results

Generate bootstrapped means

Generate 1000 mean values per bin using the spatial-temporal bootstrap for Mo.

```
Molybdenum <- temp.spat.boot(data = Mo.eux.box, var = "Mo..ppm.", reps = 1000, bin_size = bin_size, spatialScale = spatial.scale, ageScale = age.scale)
```

```
## [1] "Spatial scale is 55.5"
## [1] "Age scale is 0.0625"
## [1] "Using temporal-spatial weighting algorithm"
## [1] "No samples in 687.5 Ma bin."
## [1] "No samples in 712.5 Ma bin."
## [1] "Using temporal-spatial weighting algorithm"
## [1] "No samples in 762.5 Ma bin."
## [1] "Using temporal-spatial weighting algorithm"
## [1] "Using temporal-spatial weighting algorithm"
```

Generate 1000 mean values per bin using the spatial-temporal bootstrap for U.

```
Uranium <- temp.spat.boot(data = U.anox.box, var = "U..ppm.", reps = 1000, bin_size = bin_size, spatialScale = spatial.scale, ageScale = age.scale)
```

```
## [1] "Spatial scale is 55.5"
## [1] "Age scale is 0.0625"
## [1] "Using temporal-spatial weighting algorithm"
## [1] "No samples in 862.5 Ma bin."
## [1] "No samples in 887.5 Ma bin."
## [1] "No samples in 912.5 Ma bin."
## [1] "No samples in 937.5 Ma bin."
## [1] "Using temporal-spatial weighting algorithm"
```

Generate 1000 mean values per bin using the spatial-temporal bootstrap for proportion euxinic.

```
Eux.Fe <- temp.spat.boot(data = Fepy.anox.box, var = "euxinic.Fe", reps = 1000, bin_size = bin_size, spatialScale = spatial.scale, ageScale = age.scale)
```

```
## [1] "Spatial scale is 55.5"
## [1] "Age scale is 0.0625"
## [1] "Using temporal-spatial weighting algorithm"
## [1] "No samples in 712.5 Ma bin."
## [1] "Using temporal-spatial weighting algorithm"
## [1] "No samples in 862.5 Ma bin."
## [1] "Using temporal-spatial weighting algorithm"
## [1] "Using temporal-spatial weighting algorithm"
## [1] "No samples in 937.5 Ma bin."
## [1] "No samples in 962.5 Ma bin."
## [1] "Using temporal-spatial weighting algorithm"
```

Generate 1000 mean values per bin using the spatial-temporal bootstrap for TOC.

```
TOC <- temp.spat.boot(data = TOC.all.box, var = "TOC..wt..", reps = 1000, bin_size = bin_size, spatialScale = spatial.scale, ageScale = age.scale)
```

```
## [1] "Spatial scale is 55.5"
## [1] "Age scale is 0.0625"
## [1] "Using temporal-spatial weighting algorithm"
## [1] "No samples in 862.5 Ma bin."
## [1] "Using temporal-spatial weighting algorithm"
## [1] "No samples in 912.5 Ma bin."
## [1] "No samples in 937.5 Ma bin."
## [1] "Using temporal-spatial weighting algorithm"
## [1] "Using temporal-spatial weighting algorithm"
```

Generate box and whisker plots of bootstrapped means

Mo

```

Mo.box.standalone <- ggplot(Molybdenum, aes(x=time.bin, y=boot.mean, group=time.bin))+  

  stat_boxplot(geom = 'errorbar', size=0.6, width = 6, color="grey20", lwd=.1) +  

  geom_boxplot(outlier.alpha = .5, outlier.shape=16, outlier.size=2, outlier.color  
="grey20", width=15, color="grey20", lwd=0.6, fill="grey80", fatten = 1.3)+  

  coord_geo(xlim=c(start_age+1,end_age-1), ylim=c(-.5,225),expand=FALSE,  

            pos = as.list(rep("bottom", 1)),  

            abbrv=list( T),  

            dat = list(periods.edit),  

            height = list(unit(2, "lines")),  

            size=list(7),  

            bord=list(c("left", "bottom", "right")), lwd=as.list(1))+  

  theme_minimal() + scale_x_reverse(breaks=c(300,400,500,600,700,800,900,1000)) + #breaks=c(360,400,440,480))+  

  ylab("Mo (ppm)") + xlab("Time (Ma)") +  

  theme(plot.margin = ggplot2::margin(0,1,1,1,"cm"), panel.border = element_rect(fill=NA,color="black", size=2,linetype="solid"),  

    axis.ticks = element_line(size=1),  

    axis.line = element_line(lineend = 'square'),  

    axis.title = element_text(size=34),  

    axis.text = element_text( size=26, color="black"),  

    legend.title = element_text(size=24),  

    legend.text = element_text( size=18),  

    axis.ticks.length = unit(5, "points"),  

    legend.position="none",  

    panel.grid.major = element_blank(), panel.grid.minor = element_blank())

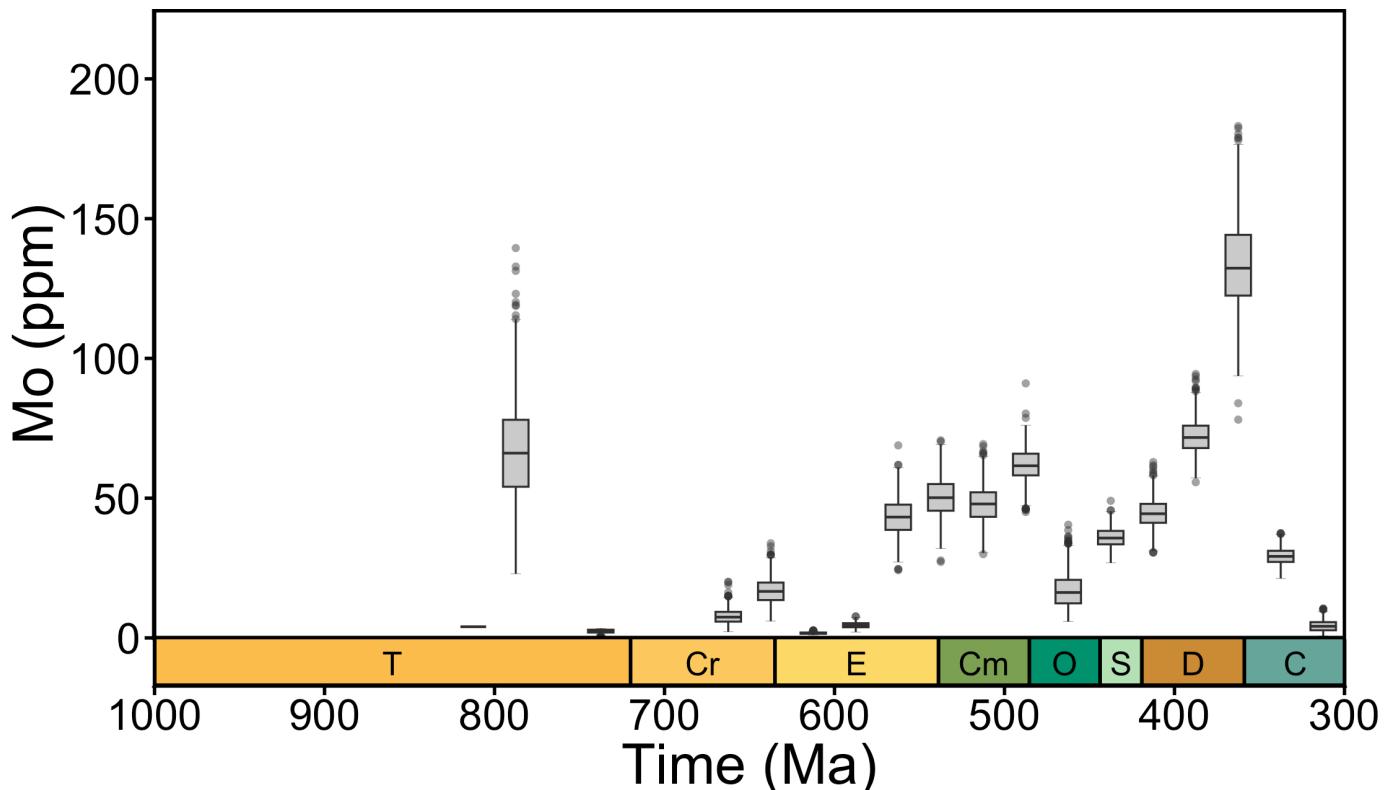
```

Mo.box.standalone

```

## Warning: Removed 3000 rows containing non-finite values (`stat_boxplot()`).
## Removed 3000 rows containing non-finite values (`stat_boxplot()`).

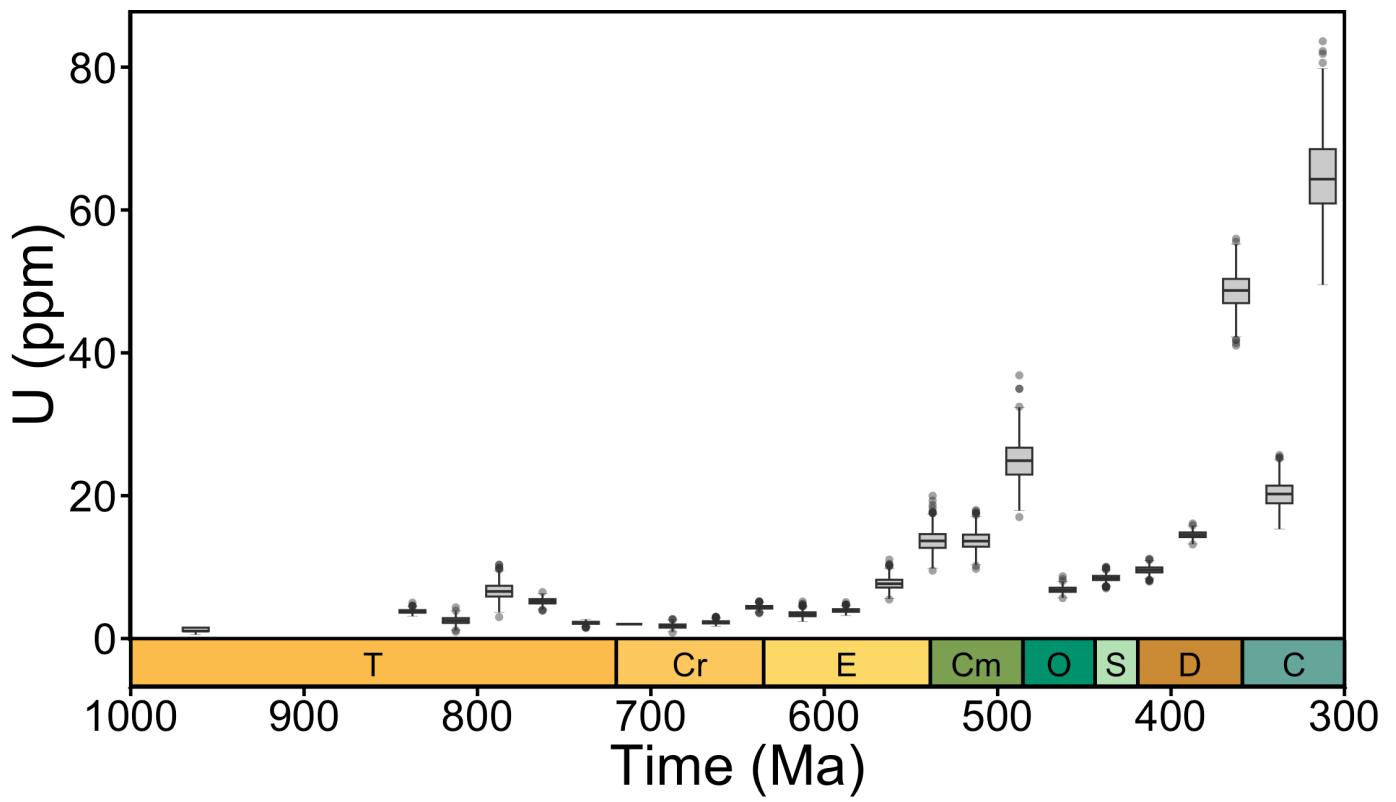
```



U

```
U.box.standalone <- ggplot(Uranium, aes(x=time.bin, y=boot.mean, group=time.bin))+  
  stat_boxplot(geom = 'errorbar', size=0.6, width = 6, color="grey20", lwd=.1) +  
  geom_boxplot(outlier.alpha = .5, outlier.shape=16, outlier.size=2, outlier.color  
="grey20", width=15, color="grey20", lwd=0.6, fill="grey80", fatten = 1.3)+  
  coord_geo(xlim=c(start_age+1,end_age-1), ylim=c(-.25,88),expand=FALSE,  
            pos = as.list(rep("bottom", 1)),  
            abbrv=list( T),  
            dat = list(periods.edit),  
            height = list(unit(2, "lines")),  
            size=list(7),  
            bord=list(c("left", "bottom", "right")), lwd=as.list(1))+  
  theme_minimal()+scale_x_reverse(breaks=c(300,400,500,600,700,800,900,1000))+#bre  
aks=c(360,400,440,480))+  
  ylab("U (ppm)")+xlab("Time (Ma)")+  
  theme(plot.margin = ggplot2::margin(0,1,1,1,"cm"),panel.border = element_rect(fi  
ll=NA,color="black", size=2,linetype="solid"),  
        axis.ticks = element_line(size=1),  
        axis.line = element_line(lineend = 'square'),  
        axis.title = element_text(size=34),  
        axis.text = element_text( size=26, color="black"),  
        legend.title = element_text(size=24),  
        legend.text = element_text( size=18),  
        axis.ticks.length = unit(5, "points"),  
        legend.position="none",  
        panel.grid.major = element_blank(),panel.grid.minor = element_blank())  
U.box.standalone
```

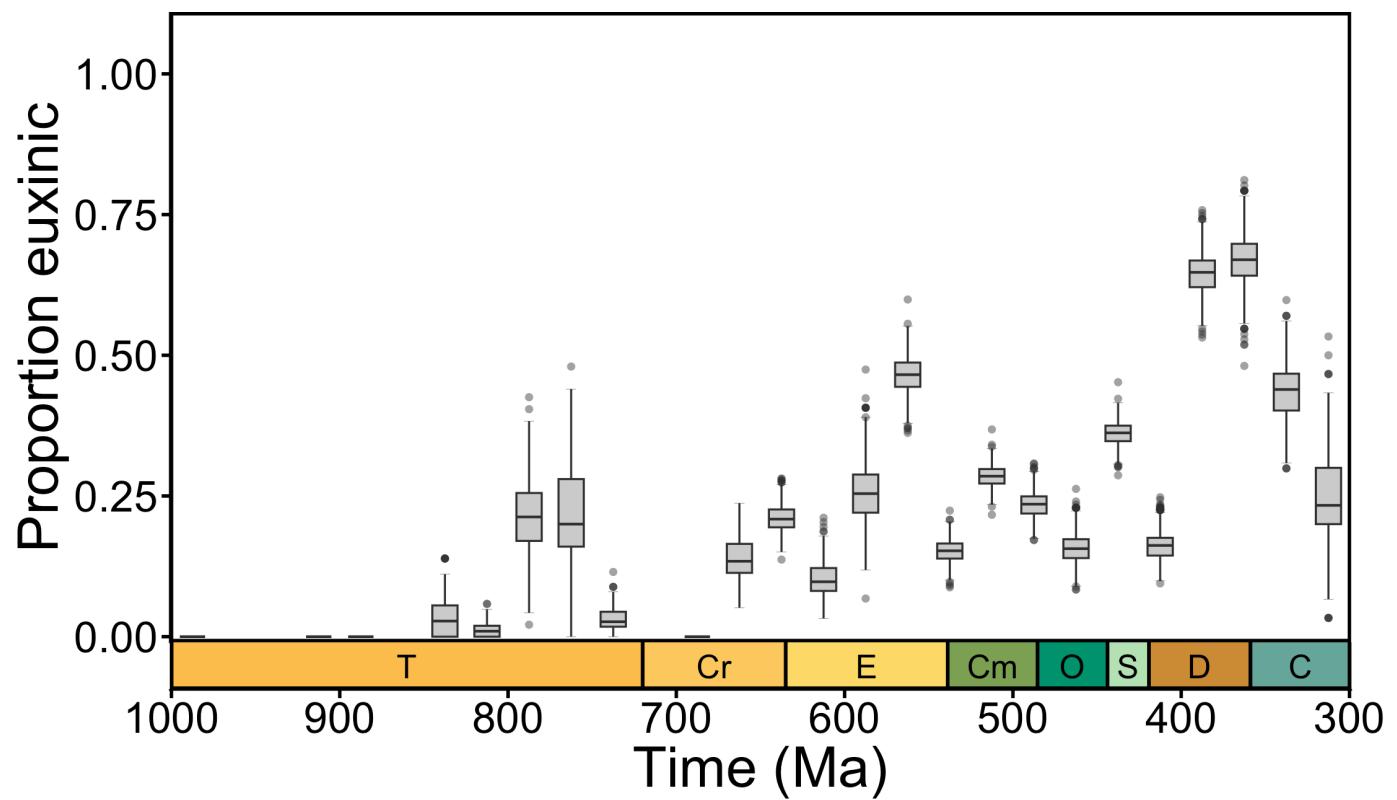
```
## Warning: Removed 4000 rows containing non-finite values (`stat_boxplot()`).  
## Removed 4000 rows containing non-finite values (`stat_boxplot()`).
```



Proportion euxinic

```
Fepy.box.standalone <- ggplot(Eux.Fe, aes(x=time.bin, y=boot.mean, group=time.bin)) +
  stat_boxplot(geom = 'errorbar', size=0.6, width = 6, color="grey20", lwd=.1) +
  geom_boxplot(outlier.alpha = .5, outlier.shape=16, outlier.size=2, outlier.color = "grey20", width=15, color="grey20", lwd=0.6, fill="grey80", fatten = 1.3) +
  coord_geo(xlim=c(start_age+1,end_age-1), ylim=c(-.01,1.11),expand=FALSE,
            pos = as.list(rep("bottom", 1)),
            abbrv=list( T),
            dat = list(periods.edit),
            height = list(unit(2, "lines")),
            size=list(7),
            bord=list(c("left", "bottom", "right")), lwd=as.list(1))+ 
  theme_minimal() + scale_x_reverse(breaks=c(300,400,500,600,700,800,900,1000)) + #breaks=c(360,400,440,480))+ 
  ylab("Proportion euxinic") + xlab("Time (Ma)")+
  theme(plot.margin = ggplot2::margin(0,1,1,1,"cm"), panel.border = element_rect(fill=NA,color="black", size=2,linetype="solid"),
        axis.ticks = element_line(size=1),
        axis.line = element_line(lineend = 'square'),
        axis.title = element_text(size=34),
        axis.text = element_text( size=26, color="black"),
        legend.title = element_text(size=24),
        legend.text = element_text( size=18),
        axis.ticks.length = unit(5, "points"),
        legend.position="none",
        panel.grid.major = element_blank(), panel.grid.minor = element_blank())
Fepy.box.standalone
```

```
## Warning: Removed 4000 rows containing non-finite values (`stat_boxplot()`).
## Removed 4000 rows containing non-finite values (`stat_boxplot()`).
```



TOC

```

TOC.box.standalone <- ggplot(TOC, aes(x=time.bin, y=boot.mean, group=time.bin))+  

  stat_boxplot(geom = 'errorbar', size=0.6, width = 6, color="grey20", lwd=.1) +  

  geom_boxplot(outlier.alpha = .5, outlier.shape=16, outlier.size=2, outlier.color  
="grey20", width=15, color="grey20", lwd=0.6, fill="grey80", fatten = 1.3)+  

  coord_geo(xlim=c(start_age+1,end_age-1), ylim=c(-.1,8.7),expand=FALSE,  

            pos = as.list(rep("bottom", 1)),  

            abbrv=list( T),  

            dat = list(periods.edit),  

            height = list(unit(2, "lines")),  

            size=list(7),  

            bord=list(c("left", "bottom", "right")), lwd=as.list(1))+  

  theme_minimal() + scale_x_reverse(breaks=c(300,400,500,600,700,800,900,1000)) + #bre  
aks=c(360,400,440,480))+  

  ylab("TOC (wt %)") + xlab("Time (Ma)") +  

  theme(plot.margin = ggplot2::margin(0,1,1,1,"cm"), panel.border = element_rect(fi  
ll=NA,color="black", size=2,linetype="solid"),  

        axis.ticks = element_line(size=1),  

        axis.line = element_line(lineend = 'square'),  

        axis.title = element_text(size=34),  

        axis.text = element_text( size=26, color="black"),  

        legend.title = element_text(size=24),  

        legend.text = element_text( size=18),  

        axis.ticks.length = unit(5, "points"),  

        legend.position="none",  

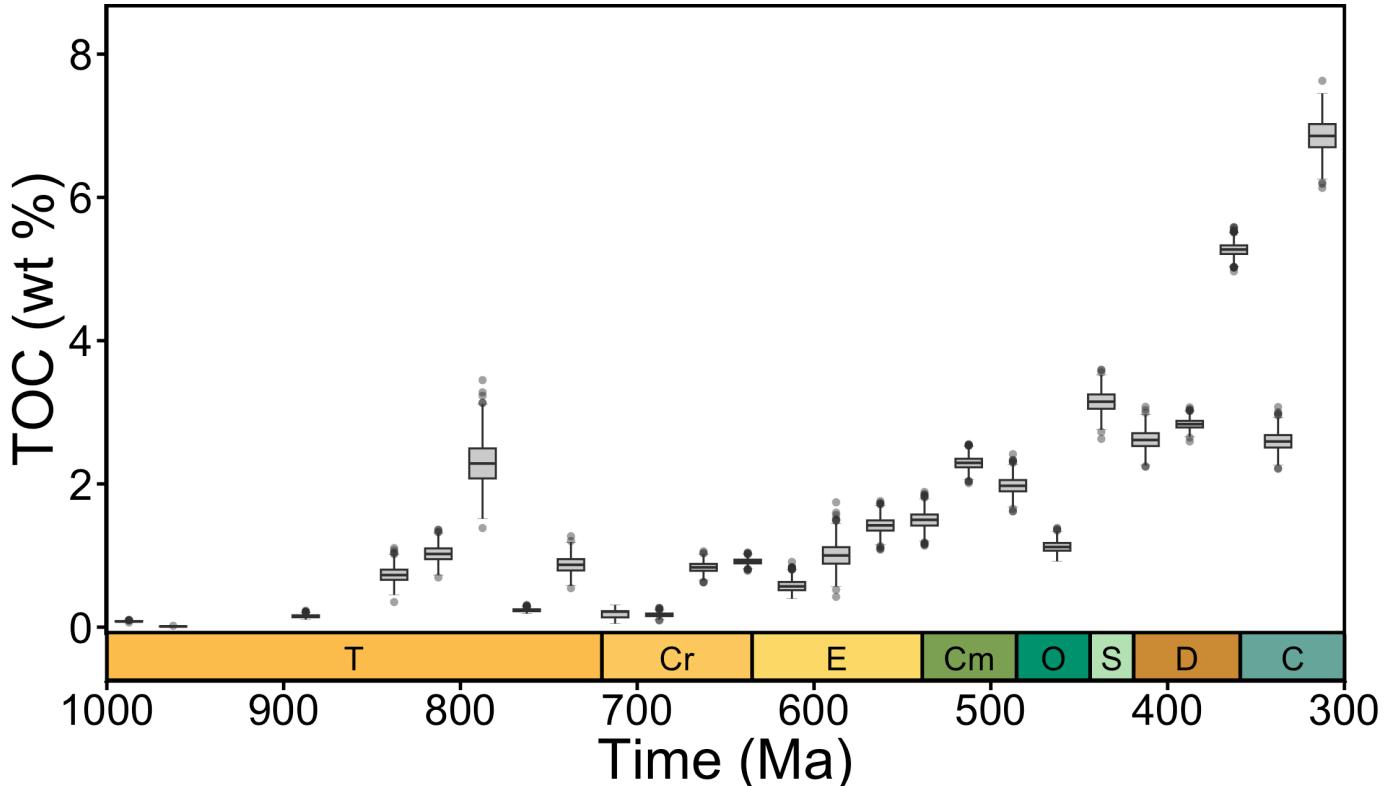
        panel.grid.major = element_blank(), panel.grid.minor = element_blank())
TOC.box.standalone

```

```

## Warning: Removed 3000 rows containing non-finite values (`stat_boxplot()`).
## Removed 3000 rows containing non-finite values (`stat_boxplot()`).

```



Plot summary figure

Define box and whisker plot panels for composite.

```
Mo.box.composite <- ggplot(Molybdenum, aes(x=time.bin, y=boot.mean, group=time.bin))+
  stat_boxplot(geom = 'errorbar', size=0.6, width = 6, color="grey20", lwd=.1) +
  geom_boxplot(outlier.alpha = .5, outlier.shape=16, outlier.size=2, outlier.color="grey20", width=15, color="grey20", lwd=0.6, fill="grey80", fatten = 1.3)+
  coord_cartesian(xlim=c(start_age+1,end_age-1), ylim=c(-.5,225),expand=FALSE)+
  theme_minimal() + scale_x_reverse(breaks=c(300,400,500,600,700,800,900,1000)) + #breaks=c(360,400,440,480))+
  ylab("Mo (ppm)") + xlab("Time (Ma)") +
  theme(plot.margin = ggplot2::margin(0,1,0,1,"cm"), panel.border = element_rect(fill=NA,color="black", size=2,linetype="solid"),
        axis.ticks = element_line(size=1),
        axis.line = element_line(lineend = 'square'),
        axis.title = element_text(size=34),
        axis.text = element_text( size=26, color="black"),
        legend.title = element_text(size=24),
        legend.text = element_text( size=18),
        axis.title.x = element_blank(),
        axis.text.x = element_blank(),
        axis.ticks.length = unit(5, "points"),
        legend.position="none",
        panel.grid.major = element_blank(), panel.grid.minor = element_blank())
```

```
U.box.composite <- ggplot(Uranium, aes(x=time.bin, y=boot.mean, group=time.bin))+
  stat_boxplot(geom = 'errorbar', size=0.6, width = 6, color="grey20", lwd=.1) +
  geom_boxplot(outlier.alpha = .5, outlier.shape=16, outlier.size=2, outlier.color="grey20", width=15, color="grey20", lwd=0.6, fill="grey80", fatten = 1.3) +
  coord_geo(xlim=c(start_age+1,end_age-1), ylim=c(-.25,88),expand=FALSE,
            pos = as.list(rep("bottom", 1)),
            abbrv=list( T),
            dat = list(periods.edit),
            height = list(unit(2, "lines")),
            size=list(7),
            bord=list(c("left", "bottom", "right")), lwd=as.list(1)) +
  theme_minimal() + scale_x_reverse(breaks=c(300,400,500,600,700,800,900,1000)) + #breaks=c(360,400,440,480)) +
  ylab("U (ppm)") + xlab("Time (Ma)") +
  theme(plot.margin = ggplot2::margin(0,1,0,1,"cm"), panel.border = element_rect(fill=NA,color="black", size=2,linetype="solid"),
        axis.ticks = element_line(size=1),
        axis.line = element_line(lineend = 'square'),
        axis.title = element_text(size=34),
        axis.text = element_text( size=26, color="black"),
        legend.title = element_text(size=24),
        legend.text = element_text( size=18),
        axis.ticks.length = unit(5, "points"),
        legend.position="none",
        panel.grid.major = element_blank(), panel.grid.minor = element_blank())
```

```

Fepy.box.composite <- ggplot(Eux.Fe, aes(x=time.bin, y=boot.mean, group=time.bin))+
  stat_boxplot(geom = 'errorbar', size=0.6, width = 6, color="grey20", lwd=.1) +
  geom_boxplot(outlier.alpha = .5, outlier.shape=16, outlier.size=2, outlier.color
="grey20", width=15, color="grey20", lwd=0.6, fill="grey80", fatten = 1.3)+ 
  coord_cartesian(c(start_age+1,end_age-1), ylim=c(-.01,1.11), expand=FALSE)+ 
  theme_minimal() + scale_x_reverse(breaks=c(300,400,500,600,700,800,900,1000)) + #breaks=c(360,400,440,480))+ 
  ylab("Proportion euxinic") + xlab("Time (Ma)") + 
  theme(plot.margin = ggplot2::margin(0,1,0,1,"cm"), panel.border = element_rect(fill=NA,color="black", size=2,linetype="solid"),
        axis.ticks = element_line(size=1),
        axis.line = element_line(lineend = 'square'),
        axis.title = element_text(size=34),
        axis.text = element_text( size=26, color="black"),
        legend.title = element_text(size=24),
        legend.text = element_text( size=18),
        axis.title.x = element_blank(),
        axis.text.x = element_blank(),
        axis.ticks.length = unit(5, "points"),
        legend.position="none",
        panel.grid.major = element_blank(), panel.grid.minor = element_blank())

TOC.box.composite <- ggplot(TOC, aes(x=time.bin, y=boot.mean, group=time.bin))+
  stat_boxplot(geom = 'errorbar', size=0.6, width = 6, color="grey20", lwd=.1) +
  geom_boxplot(outlier.alpha = .5, outlier.shape=16, outlier.size=2, outlier.color
="grey20", width=15, color="grey20", lwd=0.6, fill="grey80", fatten = 1.3)+ 
  coord_geo(xlim=c(start_age+1,end_age-1), ylim=c(-.1,8.7), expand=FALSE,
            pos = as.list(rep("bottom", 1)),
            abbrv=list( T),
            dat = list(periods.edit),
            height = list(unit(2, "lines")),
            size=list(7),
            bord=list(c("left", "bottom", "right")), lwd=as.list(1))+ 
  theme_minimal() + scale_x_reverse(breaks=c(300,400,500,600,700,800,900,1000)) + 
  ylab("TOC (wt %)") + xlab("Time (Ma)") + 
  theme(plot.margin = ggplot2::margin(0,1,1,1,"cm"), panel.border = element_rect(fill=NA,color="black", size=2,linetype="solid"),
        axis.ticks = element_line(size=1),
        axis.line = element_line(lineend = 'square'),
        axis.title = element_text(size=34),
        axis.text = element_text( size=26, color="black"),
        legend.title = element_text(size=24),
        legend.text = element_text( size=18),
        axis.ticks.length = unit(5, "points"),
        legend.position="none",
        panel.grid.major = element_blank(), panel.grid.minor = element_blank())

```

Combine 4 box and whisker plot panels and associated histograms.

```
box.fig.composite <- ggarrange2(Mo.eux.hist,
                                Fepy.anox.hist,
                                Mo.box.composite,
                                Fepy.box.composite,
                                U.anox.hist,
                                TOC.all.hist,
                                U.box.composite,
                                TOC.box.composite,
                                ncol = 2,
                                heights = c(0.2,1,0.2,1)
)
```

```
## Warning: `position_stack()` requires non-overlapping x intervals
## `position_stack()` requires non-overlapping x intervals
```

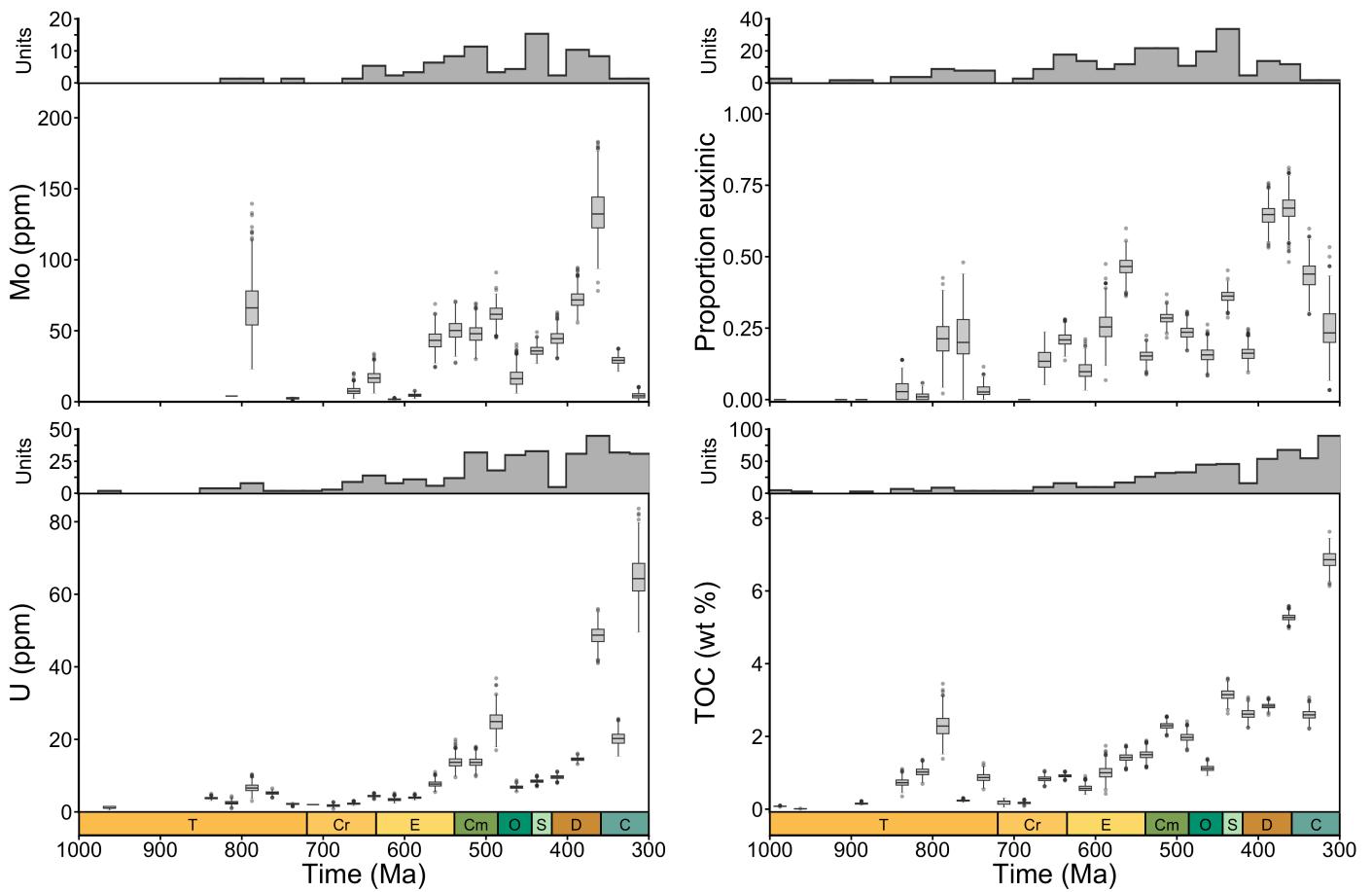
```
## Warning: Removed 3000 rows containing non-finite values (`stat_boxplot()`).
## Removed 3000 rows containing non-finite values (`stat_boxplot()`).
```

```
## Warning: Removed 4000 rows containing non-finite values (`stat_boxplot()`).
## Removed 4000 rows containing non-finite values (`stat_boxplot()`).
```

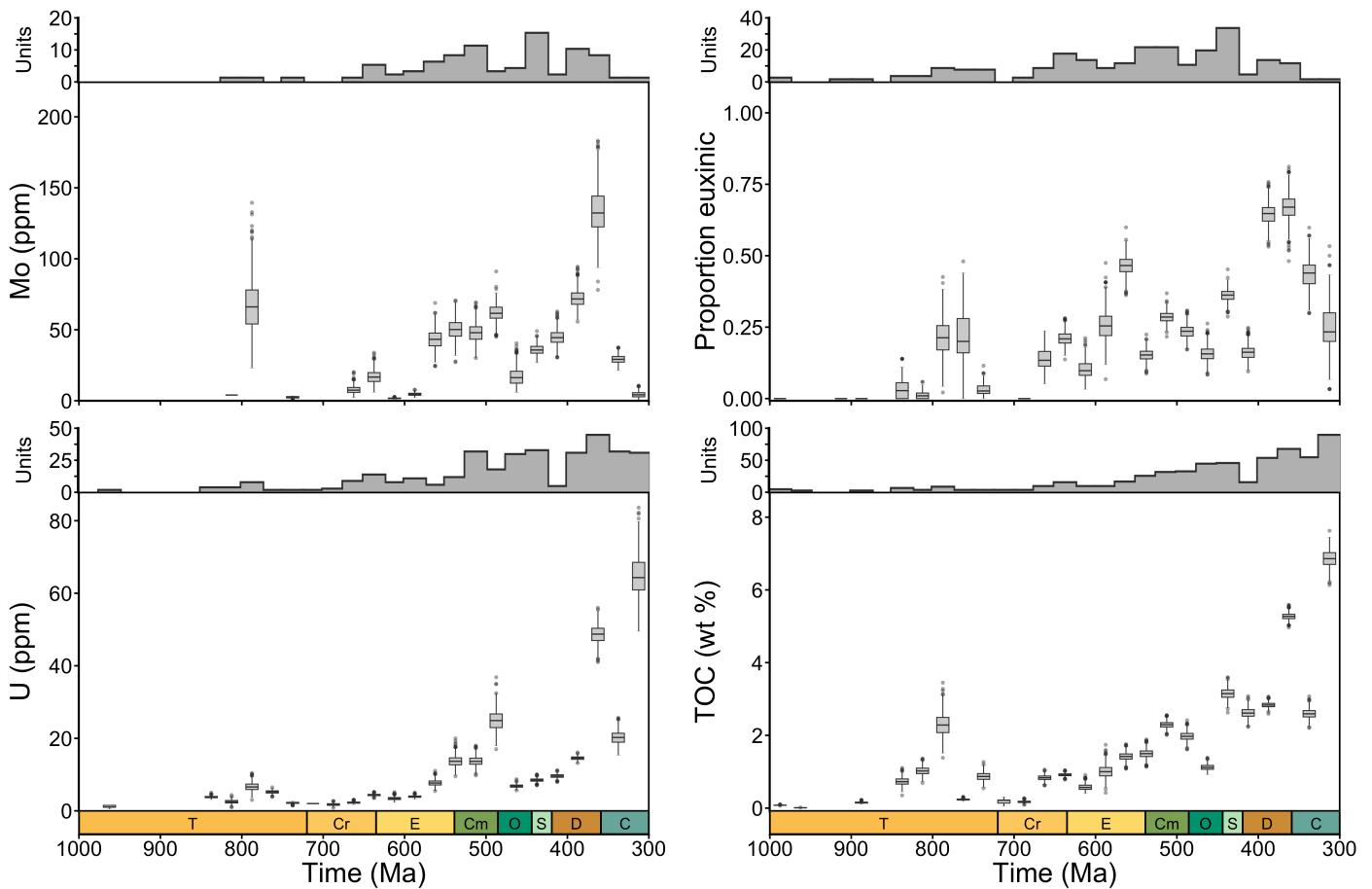
```
## Warning: `position_stack()` requires non-overlapping x intervals
## `position_stack()` requires non-overlapping x intervals
```

```
## Warning: Removed 4000 rows containing non-finite values (`stat_boxplot()`).
## Removed 4000 rows containing non-finite values (`stat_boxplot()`).
```

```
## Warning: Removed 3000 rows containing non-finite values (`stat_boxplot()`).
## Removed 3000 rows containing non-finite values (`stat_boxplot()`).
```



box.fig.composite



```
ggsave(file="Figure 1 Spatial temporal bootstrap box and whisker plots 20240208.pdf", box.fig.composite, height=16, width=24)
```

6. Bootstrapped means with no temporal-spatial weighting.

Here we generate unweighted versions of the bootstrapped means to see the impact of the temporal-spatial weighting function (which does not dramatically change the mean trends).

Generate unweighted bootstrapped means

Generate 1000 unweighted mean values per bin for Mo.

```
Molybdenum.no.weight <- temp.spat.boot(data = Mo.eux.box, var = "Mo..ppm.", reps = 1000, bin_size = bin_size, temp.spat.weight = FALSE, spatialScale = spatial.scale, ageScale = age.scale)
```

```
## [1] "Spatial scale is 55.5"  
## [1] "Age scale is 0.0625"  
## [1] "NOT using temporal-spatial weighting algorithm"  
## [1] "No samples in 687.5 Ma bin."  
## [1] "No samples in 712.5 Ma bin."  
## [1] "NOT using temporal-spatial weighting algorithm"  
## [1] "No samples in 762.5 Ma bin."  
## [1] "NOT using temporal-spatial weighting algorithm"  
## [1] "NOT using temporal-spatial weighting algorithm"
```

Generate 1000 unweighted mean values per bin for U.

```
Uranium.no.weight <- temp.spat.boot(data = U.anox.box, var = "U..ppm.", reps = 1000, bin_size = bin_size, temp.spat.weight = FALSE, spatialScale = spatial.scale, ageScale = age.scale)
```

```

## [1] "Spatial scale is 55.5"
## [1] "Age scale is 0.0625"
## [1] "NOT using temporal-spatial weighting algorithm"
## [1] "No samples in 862.5 Ma bin."
## [1] "No samples in 887.5 Ma bin."
## [1] "No samples in 912.5 Ma bin."
## [1] "No samples in 937.5 Ma bin."
## [1] "NOT using temporal-spatial weighting algorithm"

```

Generate 1000 unweighted mean values per bin for proportion euxinic.

```

Eux.Fe.no.weight <- temp.spat.boot(data = Fepy.anox.box, var = "euxinic.Fe", reps = 1000, bin_size = bin_size, temp.spat.weight = FALSE, spatialScale = spatial.scale, ageScale = age.scale)

```

```
## [1] "Spatial scale is 55.5"
## [1] "Age scale is 0.0625"
## [1] "NOT using temporal-spatial weighting algorithm"
## [1] "No samples in 712.5 Ma bin."
## [1] "NOT using temporal-spatial weighting algorithm"
## [1] "No samples in 862.5 Ma bin."
## [1] "NOT using temporal-spatial weighting algorithm"
## [1] "NOT using temporal-spatial weighting algorithm"
## [1] "No samples in 937.5 Ma bin."
## [1] "No samples in 962.5 Ma bin."
## [1] "NOT using temporal-spatial weighting algorithm"
```

Generate 1000 unweighted mean values per bin for TOC.

```
TOC.no.weight <- temp.spat.boot(data = TOC.all.box, var = "TOC..wt..", reps = 100
0, bin_size = bin_size, temp.spat.weight = FALSE, spatialScale = spatial.scale, ageScale = age.scale)
```

```
## [1] "Spatial scale is 55.5"
## [1] "Age scale is 0.0625"
## [1] "NOT using temporal-spatial weighting algorithm"
## [1] "No samples in 862.5 Ma bin."
## [1] "NOT using temporal-spatial weighting algorithm"
## [1] "No samples in 912.5 Ma bin."
## [1] "No samples in 937.5 Ma bin."
## [1] "NOT using temporal-spatial weighting algorithm"
## [1] "NOT using temporal-spatial weighting algorithm"
```

Generate box and whisker plots of bootstrapped means

Mo

```

Mo.no.weight.box.standalone <- ggplot(Molybdenum.no.weight, aes(x=time.bin, y=boot
.mean, group=time.bin))+

  stat_boxplot(geom ='errorbar', size=0.6, width = 6, color="grey20", lwd=.1) +
  geom_boxplot(outlier.alpha = .5, outlier.shape=16, outlier.size=2, outlier.color
="grey20", width=15, color="grey20", lwd=0.6, fill="grey80", fatten = 1.3)+

  coord_geo(xlim=c(start_age+1,end_age-1), ylim=c(-.5,225),expand=FALSE,
  pos = as.list(rep("bottom", 1)),
  abbrv=list( T),
  dat = list(periods.edit),
  height = list(unit(2, "lines")),
  size=list(7),
  bord=list(c("left", "bottom", "right")), lwd=as.list(1))+

  theme_minimal() + scale_x_reverse(breaks=c(300,400,500,600,700,800,900,1000)) + #bre
  aks=c(360,400,440,480))+

  ylab("Mo (ppm)") + xlab("Time (Ma)")+
  theme(plot.margin = ggplot2::margin(0,1,1,1,"cm"), panel.border = element_rect(fi
  ll=NA,color="black", size=2,linetype="solid"),
  axis.ticks = element_line(size=1),
  axis.line = element_line(lineend = 'square'),
  axis.title = element_text(size=34),
  axis.text = element_text( size=26, color="black"),
  legend.title = element_text(size=24),
  legend.text = element_text( size=18),
  axis.ticks.length = unit(5, "points"),
  legend.position="none",
  panel.grid.major = element_blank(), panel.grid.minor = element_blank())

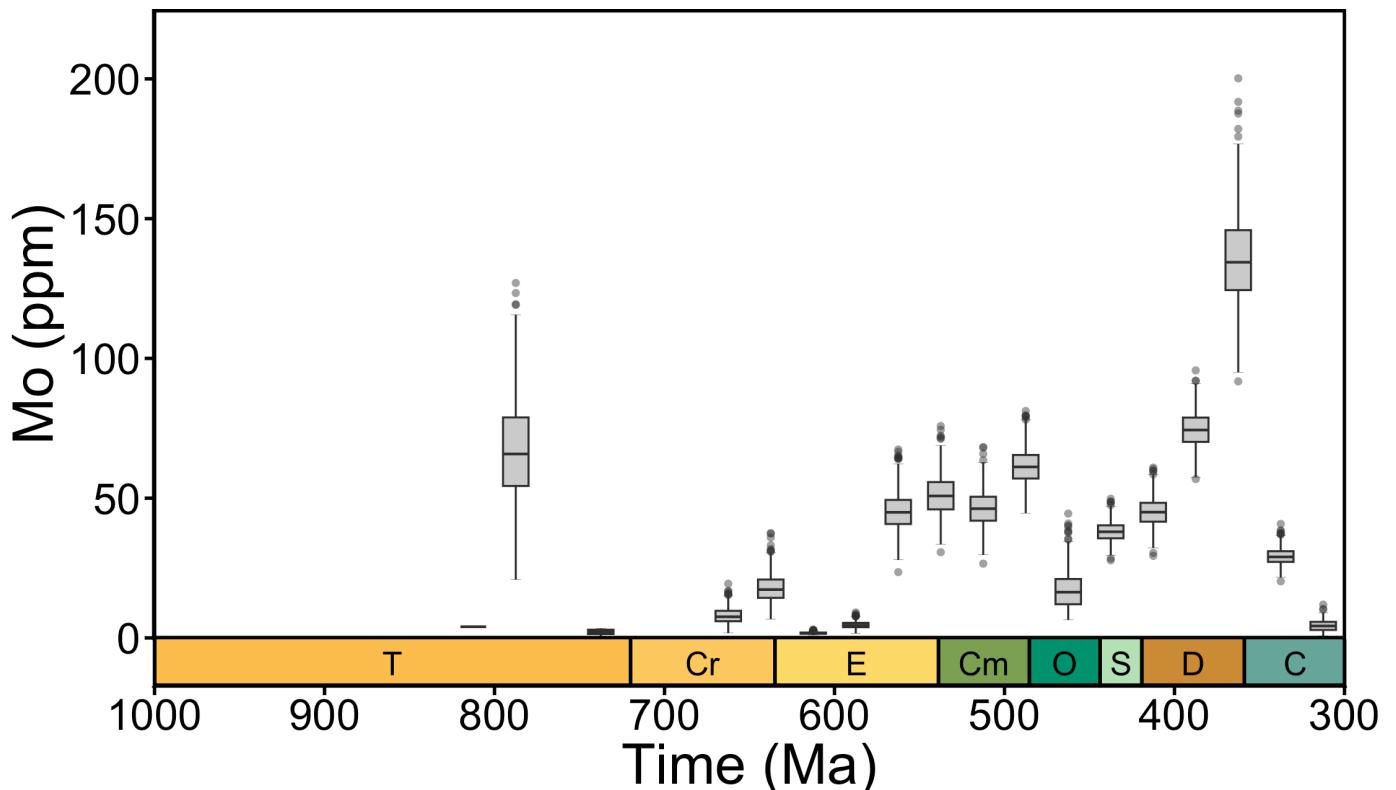
```

Mo.no.weight.box.standalone

```

## Warning: Removed 3000 rows containing non-finite values (`stat_boxplot()`).
## Removed 3000 rows containing non-finite values (`stat_boxplot()`).

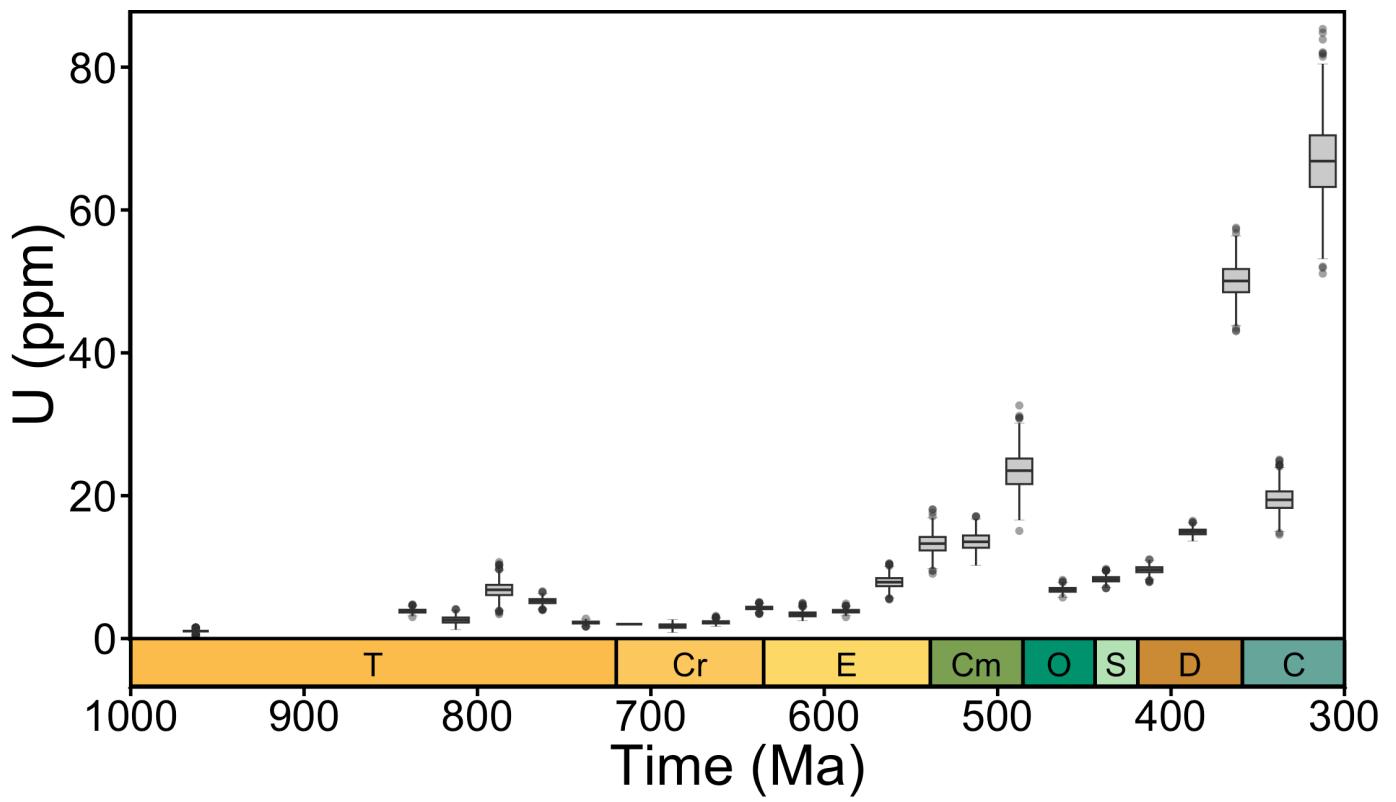
```



U

```
U.no.weight.box.standalone <- ggplot(Uranium.no.weight, aes(x=time.bin, y=boot.meau, group=time.bin))+  
  stat_boxplot(geom ='errorbar', size=0.6, width = 6, color="grey20", lwd=.1) +  
  geom_boxplot(outlier.alpha = .5, outlier.shape=16, outlier.size=2, outlier.color  
="grey20", width=15, color="grey20", lwd=0.6, fill="grey80", fatten = 1.3)+  
  coord_geo(xlim=c(start_age+1,end_age-1), ylim=c(-.25,88),expand=FALSE,  
            pos = as.list(rep("bottom", 1)),  
            abbrv=list( T),  
            dat = list(periods.edit),  
            height = list(unit(2, "lines")),  
            size=list(7),  
            bord=list(c("left", "bottom", "right")), lwd=as.list(1))+  
  theme_minimal()+scale_x_reverse(breaks=c(300,400,500,600,700,800,900,1000))+#bre  
aks=c(360,400,440,480))+  
  ylab("U (ppm)")+xlab("Time (Ma)")+  
  theme(plot.margin = ggplot2::margin(0,1,1,1,"cm"),panel.border = element_rect(fi  
ll=NA,color="black", size=2,linetype="solid"),  
        axis.ticks = element_line(size=1),  
        axis.line = element_line(lineend = 'square'),  
        axis.title = element_text(size=34),  
        axis.text = element_text( size=26, color="black"),  
        legend.title = element_text(size=24),  
        legend.text = element_text( size=18),  
        axis.ticks.length = unit(5, "points"),  
        legend.position="none",  
        panel.grid.major = element_blank(),panel.grid.minor = element_blank())  
U.no.weight.box.standalone
```

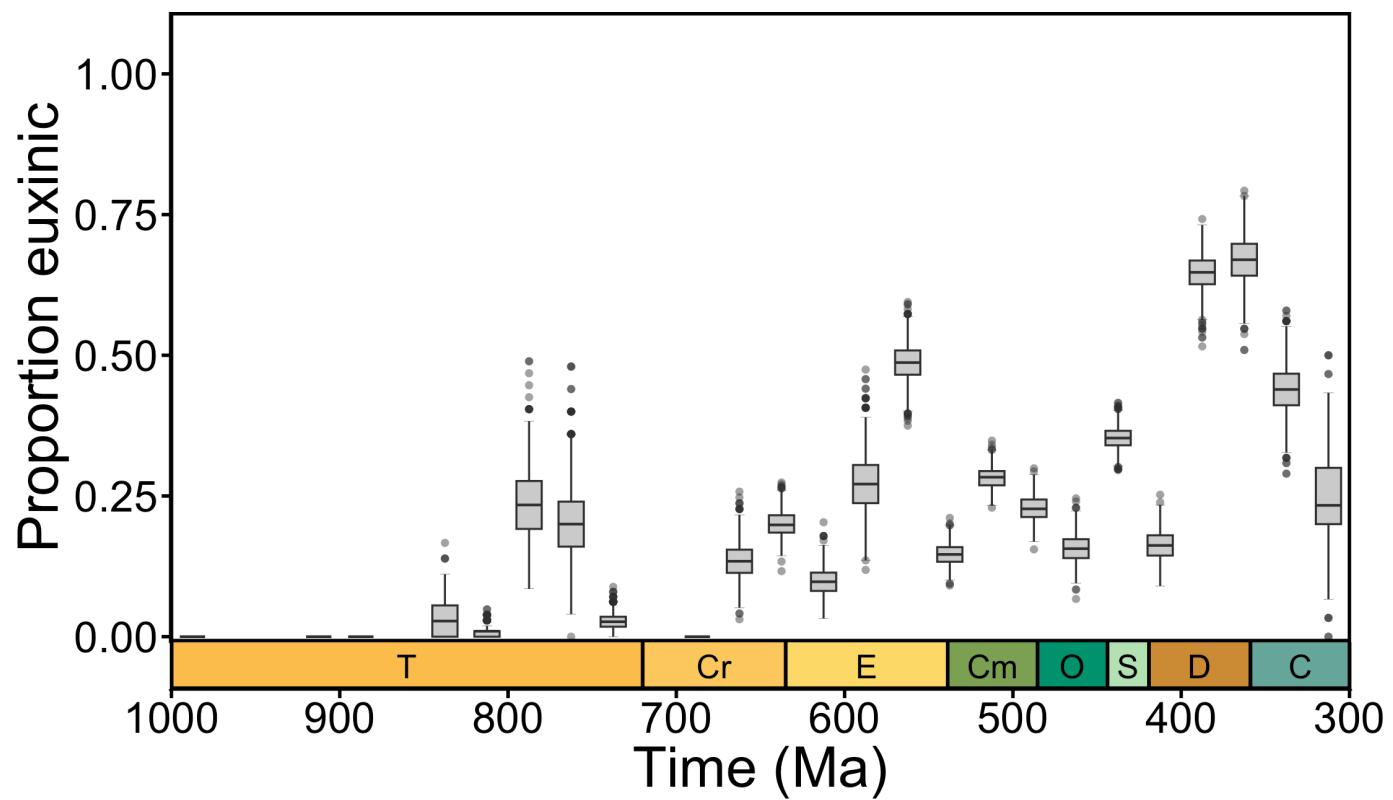
```
## Warning: Removed 4000 rows containing non-finite values (`stat_boxplot()`).  
## Removed 4000 rows containing non-finite values (`stat_boxplot()`).
```



Proportion euxinic

```
Fepy.no.weight.box.standalone <- ggplot(Eux.Fe.no.weight, aes(x=time.bin, y=boot.mean, group=time.bin))+
  stat_boxplot(geom = 'errorbar', size=0.6, width = 6, color="grey20", lwd=.1) +
  geom_boxplot(outlier.alpha = .5, outlier.shape=16, outlier.size=2, outlier.color = "grey20", width=15, color="grey20", lwd=0.6, fill="grey80", fatten = 1.3) +
  coord_geo(xlim=c(start_age+1,end_age-1), ylim=c(-.01,1.11),expand=FALSE,
            pos = as.list(rep("bottom", 1)),
            abbrv=list( T),
            dat = list(periods.edit),
            height = list(unit(2, "lines")),
            size=list(7),
            bord=list(c("left", "bottom", "right")), lwd=as.list(1))+#breaks=c(360,400,440,480))+#breaks=c(360,400,440,480))
  ylab("Proportion euxinic") + xlab("Time (Ma)")+
  theme(plot.margin = ggplot2::margin(0,1,1,1,"cm"), panel.border = element_rect(fill=NA,color="black", size=2,linetype="solid"),
        axis.ticks = element_line(size=1),
        axis.line = element_line(lineend = 'square'),
        axis.title = element_text(size=34),
        axis.text = element_text( size=26, color="black"),
        legend.title = element_text(size=24),
        legend.text = element_text( size=18),
        axis.ticks.length = unit(5, "points"),
        legend.position="none",
        panel.grid.major = element_blank(), panel.grid.minor = element_blank())
Fepy.no.weight.box.standalone
```

```
## Warning: Removed 4000 rows containing non-finite values (`stat_boxplot()`).
## Removed 4000 rows containing non-finite values (`stat_boxplot()`).
```



TOC

```

TOC.no.weight.box.standalone <- ggplot(TOC.no.weight, aes(x=time.bin, y=boot.mean,
group=time.bin))+

  stat_boxplot(geom = 'errorbar', size=0.6, width = 6, color="grey20", lwd=.1) +
  geom_boxplot(outlier.alpha = .5, outlier.shape=16, outlier.size=2, outlier.color
="grey20", width=15, color="grey20", lwd=0.6, fill="grey80", fatten = 1.3)+

  coord_geo(xlim=c(start_age+1,end_age-1), ylim=c(-.1,8.7),expand=FALSE,
  pos = as.list(rep("bottom", 1)),
  abbrv=list( T),
  dat = list(periods.edit),
  height = list(unit(2, "lines")),
  size=list(7),
  bord=list(c("left", "bottom", "right")), lwd=as.list(1))+

  theme_minimal() + scale_x_reverse(breaks=c(300,400,500,600,700,800,900,1000)) + #breaks=c(360,400,440,480))+

  ylab("TOC (wt %)")+xlab("Time (Ma)")+
  theme(plot.margin = ggplot2::margin(0,1,1,1,"cm"), panel.border = element_rect(
  fill=NA,color="black", size=2,linetype="solid"),
  axis.ticks = element_line(size=1),
  axis.line = element_line(lineend = 'square'),
  axis.title = element_text(size=34),
  axis.text = element_text( size=26, color="black"),
  legend.title = element_text(size=24),
  legend.text = element_text( size=18),
  axis.ticks.length = unit(5, "points"),
  legend.position="none",
  panel.grid.major = element_blank(), panel.grid.minor = element_blank())

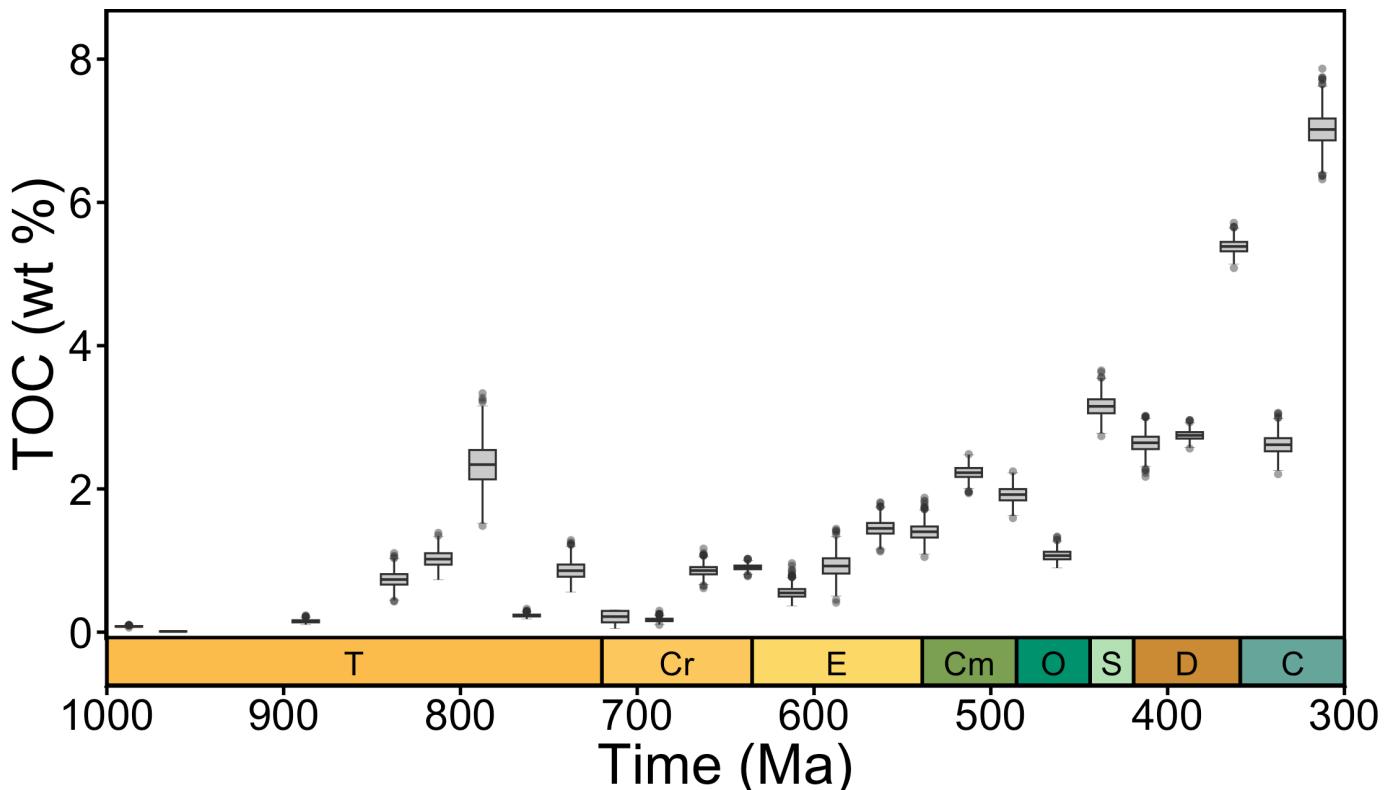
```

TOC.no.weight.box.standalone

```

## Warning: Removed 3000 rows containing non-finite values (`stat_boxplot()`).
## Removed 3000 rows containing non-finite values (`stat_boxplot()`).

```



Plot summary figure

Define box and whisker plot panels for composite.

```
Mo.no.weight.box.composite <- ggplot(Molybdenum.no.weight, aes(x=time.bin, y=boot.mean, group=time.bin))+  
  stat_boxplot(geom = 'errorbar', size=0.6, width = 6, color="grey20", lwd=.1) +  
  geom_boxplot(outlier.alpha = .5, outlier.shape=16, outlier.size=2, outlier.color  
="grey20", width=15, color="grey20", lwd=0.6, fill="grey80", fatten = 1.3)+  
  coord_cartesian(xlim=c(start_age+1,end_age-1), ylim=c(-.5,225),expand=FALSE)+  
  theme_minimal()+scale_x_reverse(breaks=c(300,400,500,600,700,800,900,1000))+#breaks=c(360,400,440,480))+  
  ylab("Mo (ppm)")+xlab("Time (Ma)")+  
  theme(plot.margin = ggplot2::margin(0,1,0,1,"cm"),panel.border = element_rect(fill=NA,color="black", size=2,linetype="solid"),  
    axis.ticks = element_line(size=1),  
    axis.line = element_line(lineend = 'square'),  
    axis.title = element_text(size=34),  
    axis.text = element_text( size=26, color="black"),  
    legend.title = element_text(size=24),  
    legend.text = element_text( size=18),  
    axis.title.x = element_blank(),  
    axis.text.x = element_blank(),  
    axis.ticks.length = unit(5, "points"),  
    legend.position="none",  
    panel.grid.major = element_blank(),panel.grid.minor = element_blank())
```

```
U.no.weight.box.composite <- ggplot(Uranium.no.weight, aes(x=time.bin, y=boot.mean  
, group=time.bin))+  
  stat_boxplot(geom = 'errorbar', size=0.6, width = 6, color="grey20", lwd=.1) +  
  geom_boxplot(outlier.alpha = .5, outlier.shape=16, outlier.size=2, outlier.color  
="grey20", width=15, color="grey20", lwd=0.6, fill="grey80", fatten = 1.3)+  
  coord_geo(xlim=c(start_age+1,end_age-1), ylim=c(-.25,88),expand=FALSE,  
    pos = as.list(rep("bottom", 1)),  
    abbrv=list( T),  
    dat = list(periods.edit),  
    height = list(unit(2, "lines")),  
    size=list(7),  
    bord=list(c("left", "bottom", "right")), lwd=as.list(1))+  
  theme_minimal()+scale_x_reverse(breaks=c(300,400,500,600,700,800,900,1000))+#breaks=c(360,400,440,480))+  
  ylab("U (ppm)")+xlab("Time (Ma)")+  
  theme(plot.margin = ggplot2::margin(0,1,0,1,"cm"),panel.border = element_rect(fill=NA,color="black", size=2,linetype="solid"),  
    axis.ticks = element_line(size=1),  
    axis.line = element_line(lineend = 'square'),  
    axis.title = element_text(size=34),  
    axis.text = element_text( size=26, color="black"),  
    legend.title = element_text(size=24),  
    legend.text = element_text( size=18),  
    axis.ticks.length = unit(5, "points"),  
    legend.position="none",
```

```

panel.grid.major = element_blank(), panel.grid.minor = element_blank()

Fepy.no.weight.box.composite <- ggplot(Eux.Fe.no.weight, aes(x=time.bin, y=boot.mean, group=time.bin))+  

  stat_boxplot(geom ='errorbar', size=0.6, width = 6, color="grey20", lwd=.1) +  

  geom_boxplot(outlier.alpha = .5, outlier.shape=16, outlier.size=2, outlier.color  
="grey20", width=15, color="grey20", lwd=0.6, fill="grey80", fatten = 1.3)+  

  coord_cartesian(c(start_age+1,end_age-1), ylim=c(-.01,1.11), expand=FALSE)+  

  theme_minimal()+scale_x_reverse(breaks=c(300,400,500,600,700,800,900,1000))+#breaks=c(360,400,440,480))+  

  ylab("Proportion euxinic")+xlab("Time (Ma)")+  

  theme(plot.margin = ggplot2::margin(0,1,0,1,"cm"),panel.border = element_rect(fi  
ll=NA,color="black", size=2,linetype="solid"),  

  axis.ticks = element_line(size=1),  

  axis.line = element_line(lineend = 'square'),  

  axis.title = element_text(size=34),  

  axis.text = element_text( size=26, color="black"),  

  legend.title = element_text(size=24),  

  legend.text = element_text( size=18),  

  axis.title.x = element_blank(),  

  axis.text.x = element_blank(),  

  axis.ticks.length = unit(5, "points"),  

  legend.position="none",  

  panel.grid.major = element_blank(),panel.grid.minor = element_blank())

TOC.no.weight.box.composite <- ggplot(TOC.no.weight, aes(x=time.bin, y=boot.mean,  
group=time.bin))+  

  stat_boxplot(geom ='errorbar', size=0.6, width = 6, color="grey20", lwd=.1) +  

  geom_boxplot(outlier.alpha = .5, outlier.shape=16, outlier.size=2, outlier.color  
="grey20", width=15, color="grey20", lwd=0.6, fill="grey80", fatten = 1.3)+  

  coord_geo(xlim=c(start_age+1,end_age-1), ylim=c(-.1,8.7),expand=FALSE,  

    pos = as.list(rep("bottom", 1)),  

    abbrv=list( T),  

    dat = list(periods.edit),  

    height = list(unit(2, "lines")),  

    size=list(7),  

    bord=list(c("left", "bottom", "right")), lwd=as.list(1))+  

  theme_minimal()+scale_x_reverse(breaks=c(300,400,500,600,700,800,900,1000))+  

  ylab("TOC (wt %)")+xlab("Time (Ma)")+  

  theme(plot.margin = ggplot2::margin(0,1,1,1,"cm"),panel.border = element_rect(fi  
ll=NA,color="black", size=2,linetype="solid"),  

  axis.ticks = element_line(size=1),  

  axis.line = element_line(lineend = 'square'),  

  axis.title = element_text(size=34),  

  axis.text = element_text( size=26, color="black"),  

  legend.title = element_text(size=24),  

  legend.text = element_text( size=18),  

  axis.ticks.length = unit(5, "points"),  

  legend.position="none",  

  panel.grid.major = element_blank(),panel.grid.minor = element_blank())

```

Combine 4 box and whisker plot panels and associated histograms.

```
box.no.weight.fig.composite <- ggarrange2(Mo.eux.hist,
                                         Fepy.anox.hist,
                                         Mo.no.weight.box.composite,
                                         Fepy.no.weight.box.composite,
                                         U.anox.hist,
                                         TOC.all.hist,
                                         U.no.weight.box.composite,
                                         TOC.no.weight.box.composite,
                                         ncol = 2,
                                         heights = c(0.2,1,0.2,1)
)
```

```
## Warning: `position_stack()` requires non-overlapping x intervals
## `position_stack()` requires non-overlapping x intervals
```

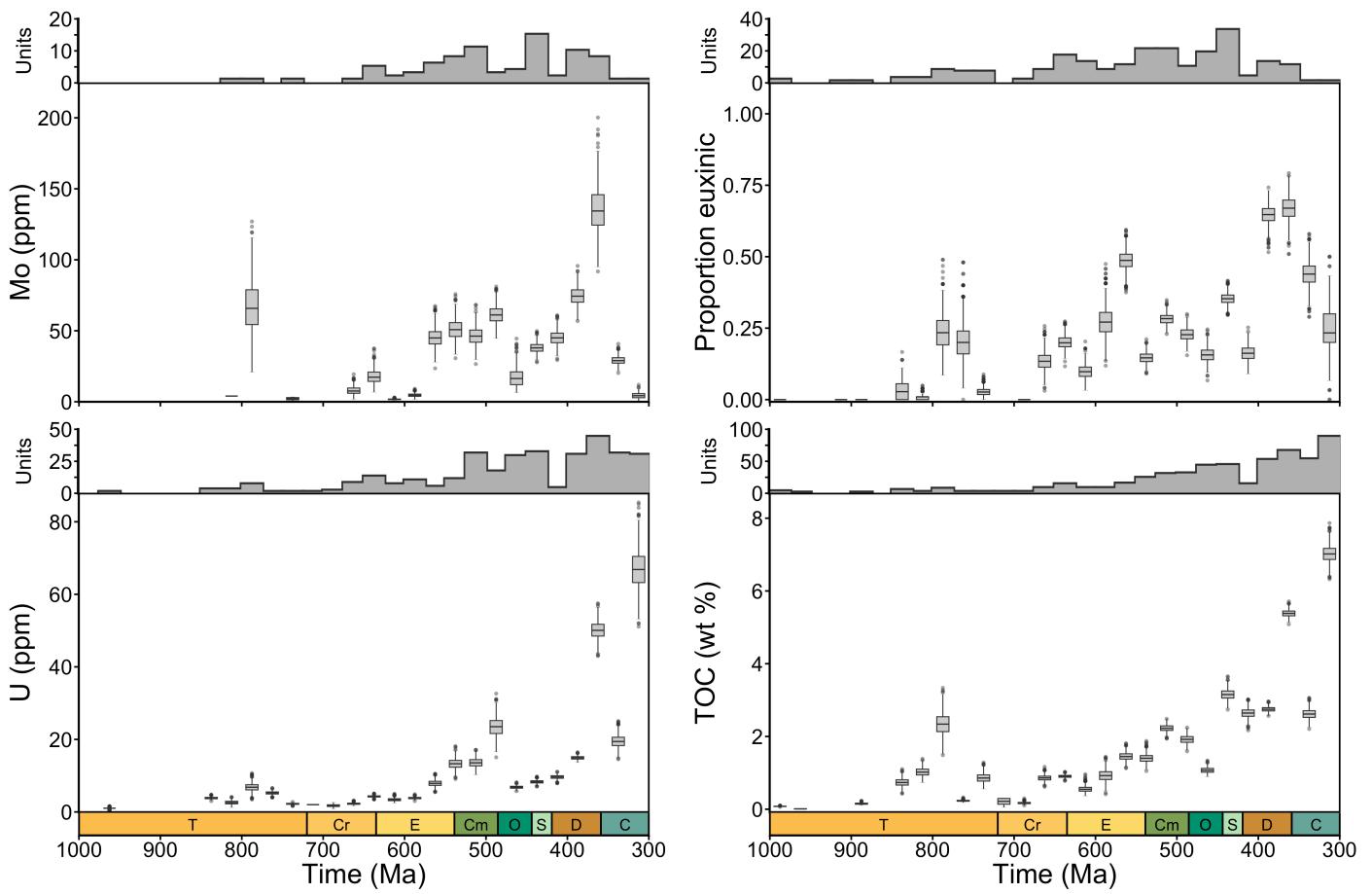
```
## Warning: Removed 3000 rows containing non-finite values (`stat_boxplot()`).
## Removed 3000 rows containing non-finite values (`stat_boxplot()`).
```

```
## Warning: Removed 4000 rows containing non-finite values (`stat_boxplot()`).
## Removed 4000 rows containing non-finite values (`stat_boxplot()`).
```

```
## Warning: `position_stack()` requires non-overlapping x intervals
## `position_stack()` requires non-overlapping x intervals
```

```
## Warning: Removed 4000 rows containing non-finite values (`stat_boxplot()`).
## Removed 4000 rows containing non-finite values (`stat_boxplot()`).
```

```
## Warning: Removed 3000 rows containing non-finite values (`stat_boxplot()`).
## Removed 3000 rows containing non-finite values (`stat_boxplot()`).
```



```
ggsave(file="Figure Sx Unweighted bootstrap box and whisker plots 20240208.pdf", b
ox.no.weight.fig.composite, height=16, width=24)
```

7. Analyses of metal/TOC ratios

Generate new subset dataframes for Mo and U relative to TOC. Exclude samples with less than 0.3% TOC to avoid generating very high ratios in low TOC samples.

Mo/TOC subdataset.

```
Mo_TOC.eux.box <- trace.toc.full %>%
  filter(!is.na(Mo..ppm.) & !is.na(TOC..wt...)) %>%
  filter(FeHR.FeT >= 0.38 & Fe.py.FeHR >= 0.7 & TOC..wt.. >= 0.3)

Mo_TOC.eux.box$Mo_TOC <- Mo_TOC.eux.box$Mo..ppm./Mo_TOC.eux.box$TOC..wt...
nrow(Mo_TOC.eux.box)
```

```
## [1] 720
```

U/TOC subdataset.

```

U_TOC.anox.box <- trace.toc.full %>%
  filter(!is.na(U..ppm.) & !is.na(TOC..wt..)) %>%
  filter((FeHR.FeT >= 0.38 | FeT.Al >= 0.53) & TOC..wt.. >= 0.3)

U_TOC.anox.box$U_TOC <- U_TOC.anox.box$U..ppm./U_TOC.anox.box$TOC..wt..

nrow(U_TOC.anox.box)

```

```
## [1] 3003
```

Bin samples.

```

Mo_TOC.eux.box$time.bin <- seq(end_age, start_age, bin_size)[as.numeric(cut(Mo_TOC.eux.box$interpreted.age, seq(end_age, start_age, bin_size)))]+bin_size/2

U_TOC.anox.box$time.bin <- seq(end_age, start_age, bin_size)[as.numeric(cut(U_TOC.anox.box$interpreted.age, seq(end_age, start_age, bin_size)))]+bin_size/2

```

Generate 1000 mean values per bin using the spatial-temporal bootstrap for Mo/TOC.

```
Molybdenum_TOC <- temp.spat.boot(data = Mo_TOC.eux.box, var = "Mo_TOC", reps = 1000, bin_size = bin_size, spatialScale = spatial.scale, ageScale = age.scale)
```

```

## [1] "Spatial scale is 55.5"
## [1] "Age scale is 0.0625"
## [1] "Using temporal-spatial weighting algorithm"
## [1] "No samples in 687.5 Ma bin."
## [1] "No samples in 712.5 Ma bin."
## [1] "No samples in 737.5 Ma bin."
## [1] "No samples in 762.5 Ma bin."
## [1] "Using temporal-spatial weighting algorithm"
## [1] "Using temporal-spatial weighting algorithm"

```

Generate 1000 mean values per bin using the spatial-temporal bootstrap for U/TOC.

```
Uranium_TOC <- temp.spat.boot(data = U_TOC.anox.box, var = "U_TOC", reps = 1000, bin_size = bin_size, spatialScale = spatial.scale, ageScale = age.scale)
```

```
## [1] "Spatial scale is 55.5"  
## [1] "Age scale is 0.0625"  
## [1] "Using temporal-spatial weighting algorithm"  
## [1] "No samples in 762.5 Ma bin."  
## [1] "Using temporal-spatial weighting algorithm"  
## [1] "Using temporal-spatial weighting algorithm"  
## [1] "Using temporal-spatial weighting algorithm"
```

Mo/TOC standalone.

```

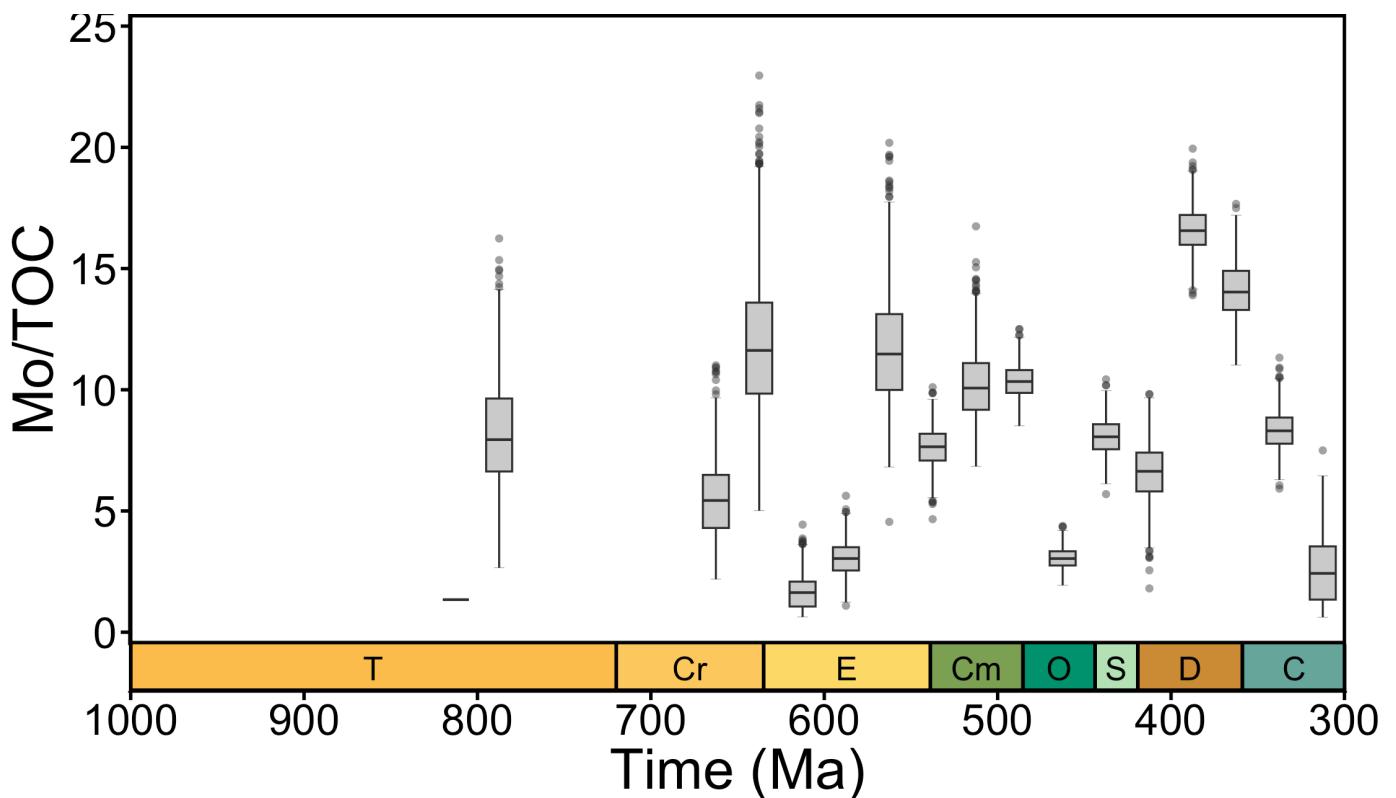
Mo_TOC.box.standalone <- ggplot(Molybdenum_TOC, aes(x=time.bin, y=boot.mean, group =time.bin))+ 
  stat_boxplot(geom = 'errorbar', size=0.6, width = 6, color="grey20", lwd=.1) + 
  geom_boxplot(outlier.alpha = .5, outlier.shape=16, outlier.size=2, outlier.color ="grey20", width=15, color="grey20", lwd=0.6, fill="grey80", fatten = 1.3)+ 
  coord_geo(xlim=c(start_age+1,end_age-1), ylim=c(-.5,25.5),expand=FALSE, 
             pos = as.list(rep("bottom", 1)), 
             abbrv=list( T),
             dat = list(periods.edit),
             height = list(unit(2, "lines")),
             size=list(7),
             bord=list(c("left", "bottom", "right")), lwd=as.list(1))+ 
  theme_minimal() + scale_x_reverse(breaks=c(300,400,500,600,700,800,900,1000)) + #breaks=c(360,400,440,480))+ 
  ylab("Mo/TOC") + xlab("Time (Ma)")+ 
  theme(plot.margin = ggplot2::margin(0,1,1,1,"cm"), panel.border = element_rect(fill=NA,color="black", size=2,linetype="solid"),
        axis.ticks = element_line(size=1),
        axis.line = element_line(lineend = 'square'),
        axis.title = element_text(size=34),
        axis.text = element_text( size=26, color="black"),
        legend.title = element_text(size=24),
        legend.text = element_text( size=18),
        axis.ticks.length = unit(5, "points"),
        legend.position="none",
        panel.grid.major = element_blank(), panel.grid.minor = element_blank())
Mo_TOC.box.standalone

```

```

## Warning: Removed 4000 rows containing non-finite values (`stat_boxplot()`).
## Removed 4000 rows containing non-finite values (`stat_boxplot()`).

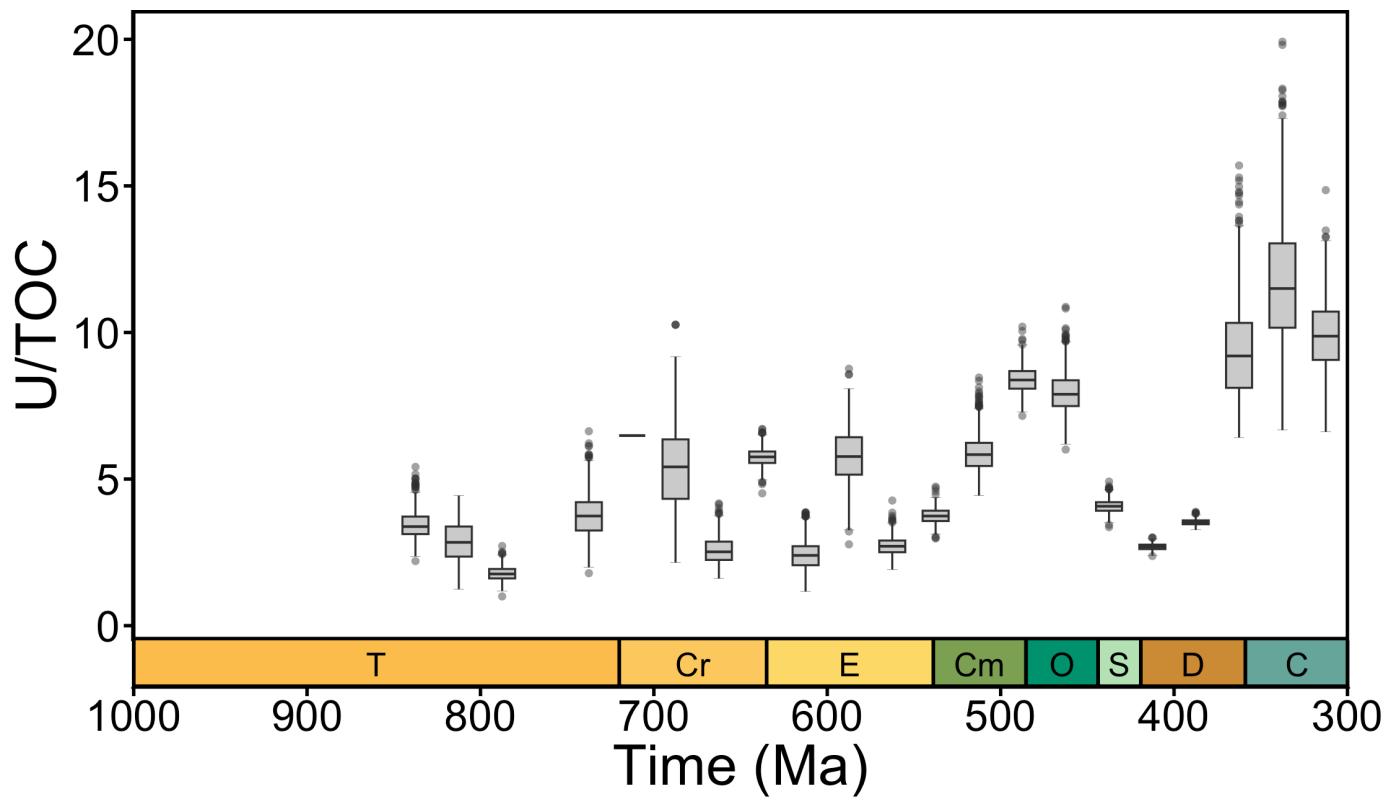
```



U/TOC standalone.

```
U_TOC.box.standalone <- ggplot(Uranium_TOC, aes(x=time.bin, y=boot.mean, group=time.bin))+
  stat_boxplot(geom ='errorbar', size=0.6, width = 6, color="grey20", lwd=.1) +
  geom_boxplot(outlier.alpha = .5, outlier.shape=16, outlier.size=2, outlier.color ="grey20", width=15, color="grey20", lwd=0.6, fill="grey80", fatten = 1.3) +
  coord_geo(xlim=c(start_age+1,end_age-1), ylim=c(-.5,21),expand=FALSE,
             pos = as.list(rep("bottom", 1)),
             abbrv=list( T),
             dat = list(periods.edit),
             height = list(unit(2, "lines")),
             size=list(7),
             bord=list(c("left", "bottom", "right")), lwd=as.list(1))+theme_minimal() + scale_x_reverse(breaks=c(300,400,500,600,700,800,900,1000)) + #breaks=c(360,400,440,480)) +
  ylab("U/TOC") + xlab("Time (Ma)") +
  theme(plot.margin = ggplot2::margin(0,1,1,1,"cm"), panel.border = element_rect(fill=NA,color="black", size=2,linetype="solid"),
        axis.ticks = element_line(size=1),
        axis.line = element_line(lineend = 'square'),
        axis.title = element_text(size=34),
        axis.text = element_text( size=26, color="black"),
        legend.title = element_text(size=24),
        legend.text = element_text( size=18),
        axis.ticks.length = unit(5, "points"),
        legend.position="none",
        panel.grid.major = element_blank(), panel.grid.minor = element_blank())
U_TOC.box.standalone
```

```
## Warning: Removed 1000 rows containing non-finite values (`stat_boxplot()`).
## Removed 1000 rows containing non-finite values (`stat_boxplot()`).
```



Generate histograms for metal/TOC composite figure.

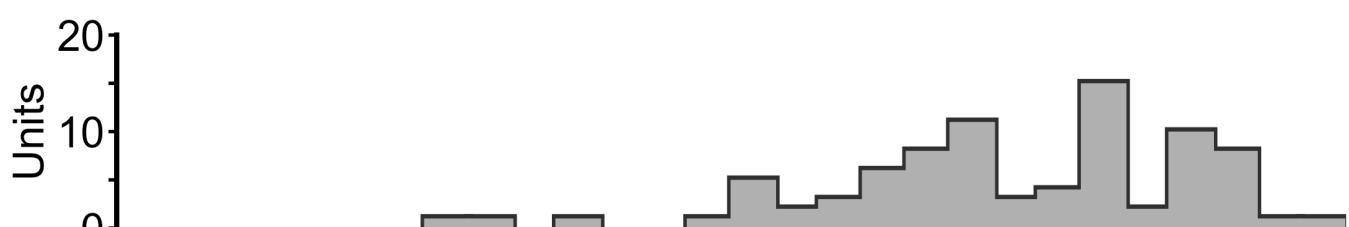
```

Mo_TOC.eux.units <- Mo_TOC.eux.box %>%
  group_by(long.stratigraphy.name, time.bin) %>%
  tally() %>%
  group_by(time.bin) %>%
  as.data.frame()

Mo_TOC.eux.hist <- ggplot(Mo_TOC.eux.units, aes(time.bin)) +
  geom_bar(fill="grey70", color="grey20", size=2.5, width=25) +
  geom_bar(fill="grey70", color="grey70", size=0, width=26) +
  coord_cartesian(xlim=c(start_age+1,end_age-1), ylim=c(0,20), expand=FALSE) +
  theme_minimal() + scale_x_reverse(breaks=c(300,400,500,600,700,800,900,1000)) + #breaks=c(360,400,440,480)) +
  scale_y_continuous(breaks=c(0,5,10,15,20), labels=c("0","",""10,"",""20")) +
  ylab("Units") +
  xlab("Time (Ma)") +
  theme(plot.margin = ggplot2::margin(1,1,-0.2,1,"cm"), panel.border = element_rect(
    fill=NA, color=NA, size=2, linetype="solid"),
    axis.line.x = element_blank(),
    axis.line.y = element_line(lineend = 'square', color="black", size=1.5),
    axis.ticks = element_line(size=1),
    axis.ticks.x = element_blank(),
    axis.title = element_text(size=26),
    axis.text = element_text( size=26, color="black"),
    legend.text = element_text( size=16),
    legend.title = element_text( size=16),
    legend.position="none",
    legend.background = element_rect(color=NA, fill=NA),
    axis.title.x = element_blank(),
    axis.text.x = element_blank(),
    axis.ticks.length = unit(5, "points"),
    panel.grid.major = element_blank(), panel.grid.minor = element_blank())

```

Warning: `position_stack()` requires non-overlapping x intervals



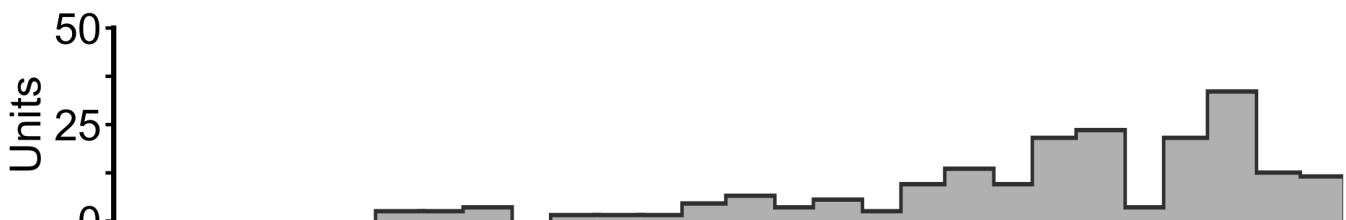
```

U_TOC.anox.units <- U_TOC.anox.box %>%
  group_by(long.stratigraphy.name, time.bin) %>%
  tally() %>%
  group_by(time.bin) %>%
  as.data.frame()

U_TOC.anox.hist <- ggplot(U_TOC.anox.units, aes(time.bin))+
  geom_bar(fill="grey70", color="grey20", size=2.5, width=25)+
  geom_bar(fill="grey70", color="grey70", size=0, width=26)+
  coord_cartesian(xlim=c(start_age+1,end_age-1), ylim=c(0,50), expand=FALSE)+
  theme_minimal()+scale_x_reverse(breaks=c(300,400,500,600,700,800,900,1000))+#breaks=c(360,400,440,480))+#breaks=c(360,400,440,480))
  scale_y_continuous(breaks=c(0,12.5,25,37.5,50), labels=c("0","","25","","", "50"))
+
  ylab("Units")+
  xlab("Time (Ma)")+
  theme(plot.margin = ggplot2::margin(1,1,-0.2,1,"cm"), panel.border = element_rect(
    fill=NA,color=NA, size=2,linetype="solid"),
    axis.line.x = element_blank(),
    axis.line.y = element_line(lineend = 'square', color="black", size=1.5),
    axis.ticks = element_line(size=1),
    axis.ticks.x = element_blank(),
    axis.title = element_text(size=26),
    axis.text = element_text( size=26, color="black"),
    legend.text = element_text( size=16),
    legend.title = element_text( size=16),
    legend.position="none",
    legend.background = element_rect(color=NA, fill=NA),
    axis.title.x = element_blank(),
    axis.text.x = element_blank(),
    axis.ticks.length = unit(5, "points"),
    panel.grid.major = element_blank(),panel.grid.minor = element_blank())
U_TOC.anox.hist

```

Warning: `position_stack()` requires non-overlapping x intervals



Generate panels for metal/TOC composite figure.

```

Mo_TOC.box.composite <- ggplot(Molybdenum_TOC, aes(x=time.bin, y=boot.mean, group=
time.bin))+

  stat_boxplot(geom = 'errorbar', size=0.6, width = 6, color="grey20", lwd=.1) +
  geom_boxplot(outlier.alpha = .5, outlier.shape=16, outlier.size=2, outlier.color
="grey20", width=15, color="grey20", lwd=0.6, fill="grey80", fatten = 1.3)+
  coord_cartesian(xlim=c(start_age+1,end_age-1), ylim=c(-.5,26.6),expand=FALSE)+
  theme_minimal()+scale_x_reverse(breaks=c(300,400,500,600,700,800,900,1000))+#bre
aks=c(360,400,440,480))+

  ylab("Mo/TOC")+xlab("Time (Ma)")+
  theme(plot.margin = ggplot2::margin(0,1,0,1,"cm"),panel.border = element_rect(fi
ll=NA,color="black", size=2,linetype="solid")),

  axis.ticks = element_line(size=1),
  axis.line = element_line(lineend = 'square'),
  axis.title = element_text(size=34),
  axis.text = element_text( size=26, color="black"),
  legend.title = element_text(size=24),
  legend.text = element_text( size=18),
  axis.title.x = element_blank(),
  axis.text.x = element_blank(),
  axis.ticks.length = unit(5, "points"),
  legend.position="none",
  panel.grid.major = element_blank(),panel.grid.minor = element_blank())

```

```

U_TOC.box.composite <- ggplot(Uranium_TOC, aes(x=time.bin, y=boot.mean, group=time
.bin))+

  stat_boxplot(geom = 'errorbar', size=0.6, width = 6, color="grey20", lwd=.1) +
  geom_boxplot(outlier.alpha = .5, outlier.shape=16, outlier.size=2, outlier.color
="grey20", width=15, color="grey20", lwd=0.6, fill="grey80", fatten = 1.3)+
  coord_geo(xlim=c(start_age+1,end_age-1), ylim=c(-.5,22),expand=FALSE,
  pos = as.list(rep("bottom", 1)),
  abbrv=list( T),
  dat = list(periods.edit),
  height = list(unit(2, "lines")),
  size=list(7),
  bord=list(c("left", "bottom", "right")), lwd=as.list(1))+

  theme_minimal()+scale_x_reverse(breaks=c(300,400,500,600,700,800,900,1000))+#bre
aks=c(360,400,440,480))+

  ylab("U/TOC")+xlab("Time (Ma)")+
  theme(plot.margin = ggplot2::margin(0,1,0,1,"cm"),panel.border = element_rect(fi
ll=NA,color="black", size=2,linetype="solid")),

  axis.ticks = element_line(size=1),
  axis.line = element_line(lineend = 'square'),
  axis.title = element_text(size=34),
  axis.text = element_text( size=26, color="black"),
  legend.title = element_text(size=24),
  legend.text = element_text( size=18),
  axis.ticks.length = unit(5, "points"),
  legend.position="none",
  panel.grid.major = element_blank(),panel.grid.minor = element_blank())

```

Combine 4 box and whisker plot panels and associated histograms.

```
TOC_rat.box.fig.composite <- ggarrange2(Mo_TOC.eux.hist,
                                         Fepy.anox.hist,
                                         Mo_TOC.box.composite,
                                         Fepy.box.composite,
                                         U_TOC.anox.hist,
                                         TOC.all.hist,
                                         U_TOC.box.composite,
                                         TOC.box.composite,
                                         ncol = 2,
                                         heights = c(0.2,1,0.2,1)
)
```

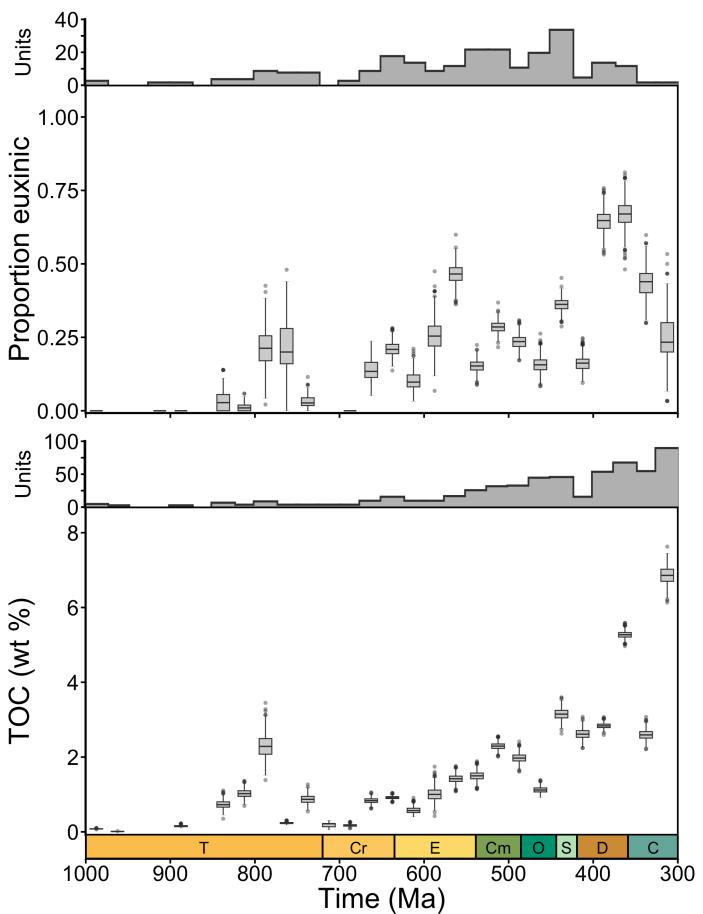
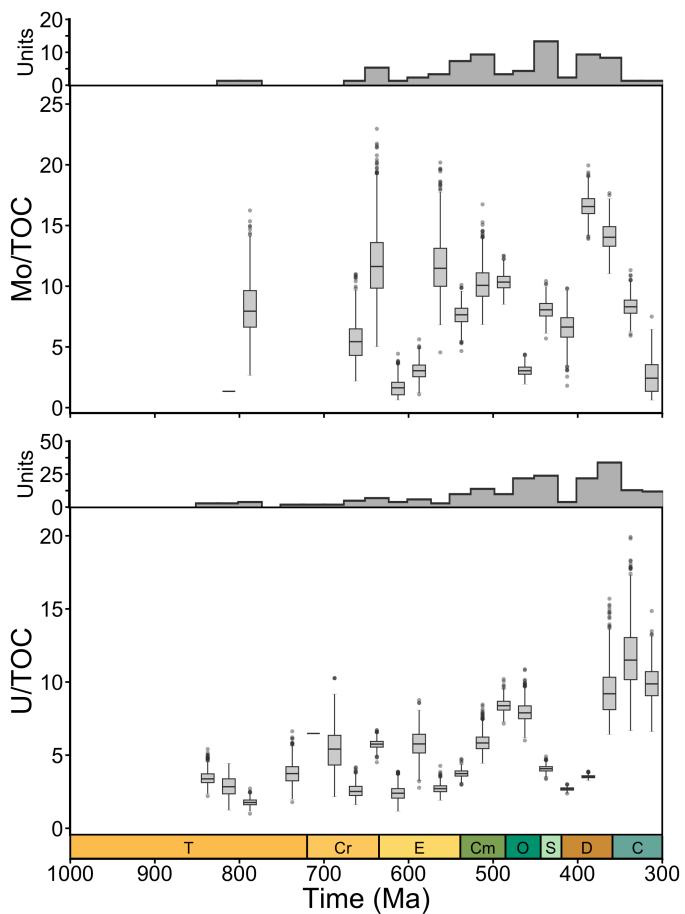
```
## Warning: `position_stack()` requires non-overlapping x intervals
## `position_stack()` requires non-overlapping x intervals
```

```
## Warning: Removed 4000 rows containing non-finite values (`stat_boxplot()`).
## Removed 4000 rows containing non-finite values (`stat_boxplot()`).
## Removed 4000 rows containing non-finite values (`stat_boxplot()`).
## Removed 4000 rows containing non-finite values (`stat_boxplot()`).
```

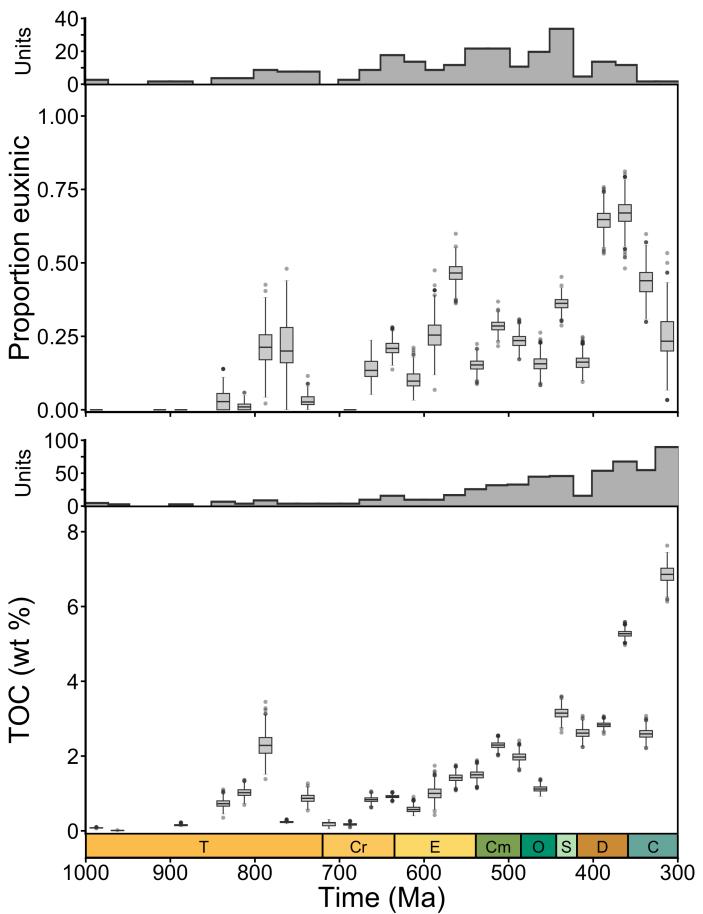
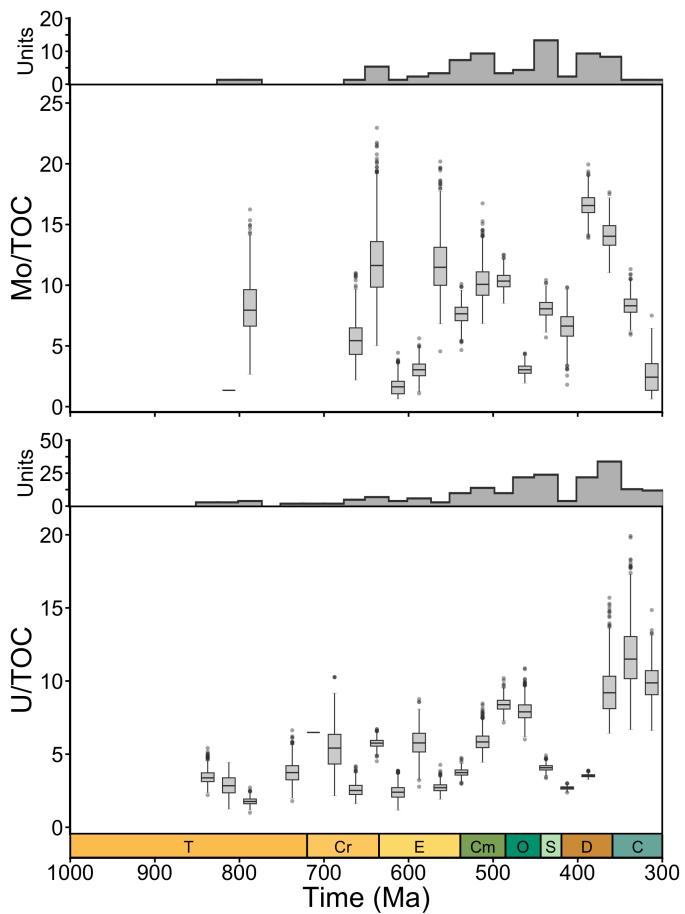
```
## Warning: `position_stack()` requires non-overlapping x intervals
## `position_stack()` requires non-overlapping x intervals
```

```
## Warning: Removed 1000 rows containing non-finite values (`stat_boxplot()`).
## Removed 1000 rows containing non-finite values (`stat_boxplot()`).
```

```
## Warning: Removed 3000 rows containing non-finite values (`stat_boxplot()`).
## Removed 3000 rows containing non-finite values (`stat_boxplot()`).
```



TOC_rat.box.fig.composite



```
ggsave(file="Figure Sx Spatial temporal bootstrap box and whisker plots (metal-TOC ratios) 20240208.pdf", TOC_rat.box.fig.composite, height=16, width=24)
```

7b. New Figure 1...

Now - based on reviewer feedback we are going to build a 6 panel version of Figure 1, combining old figure 1 and old Extended Figure 3.

Combine 4 box and whisker plot panels and associated histograms.

```
box.fig.composite <- ggarrange2(Mo.eux.hist,
                                Mo_TOC.eux.hist,
                                Fepy.anox.hist,
                                Mo.box.composite,
                                Mo_TOC.box.composite,
                                Fepy.box.composite,
                                U.anox.hist,
                                U_TOC.anox.hist,
                                TOC.all.hist,
                                U.box.composite,
                                U_TOC.box.composite,
                                TOC.box.composite,
                                ncol = 3,
                                heights = c(0.2,1,0.2,1)
)
```

```
## Warning: `position_stack()` requires non-overlapping x intervals
## `position_stack()` requires non-overlapping x intervals
## `position_stack()` requires non-overlapping x intervals
```

```
## Warning: Removed 3000 rows containing non-finite values (`stat_boxplot()`).
## Removed 3000 rows containing non-finite values (`stat_boxplot()`).
```

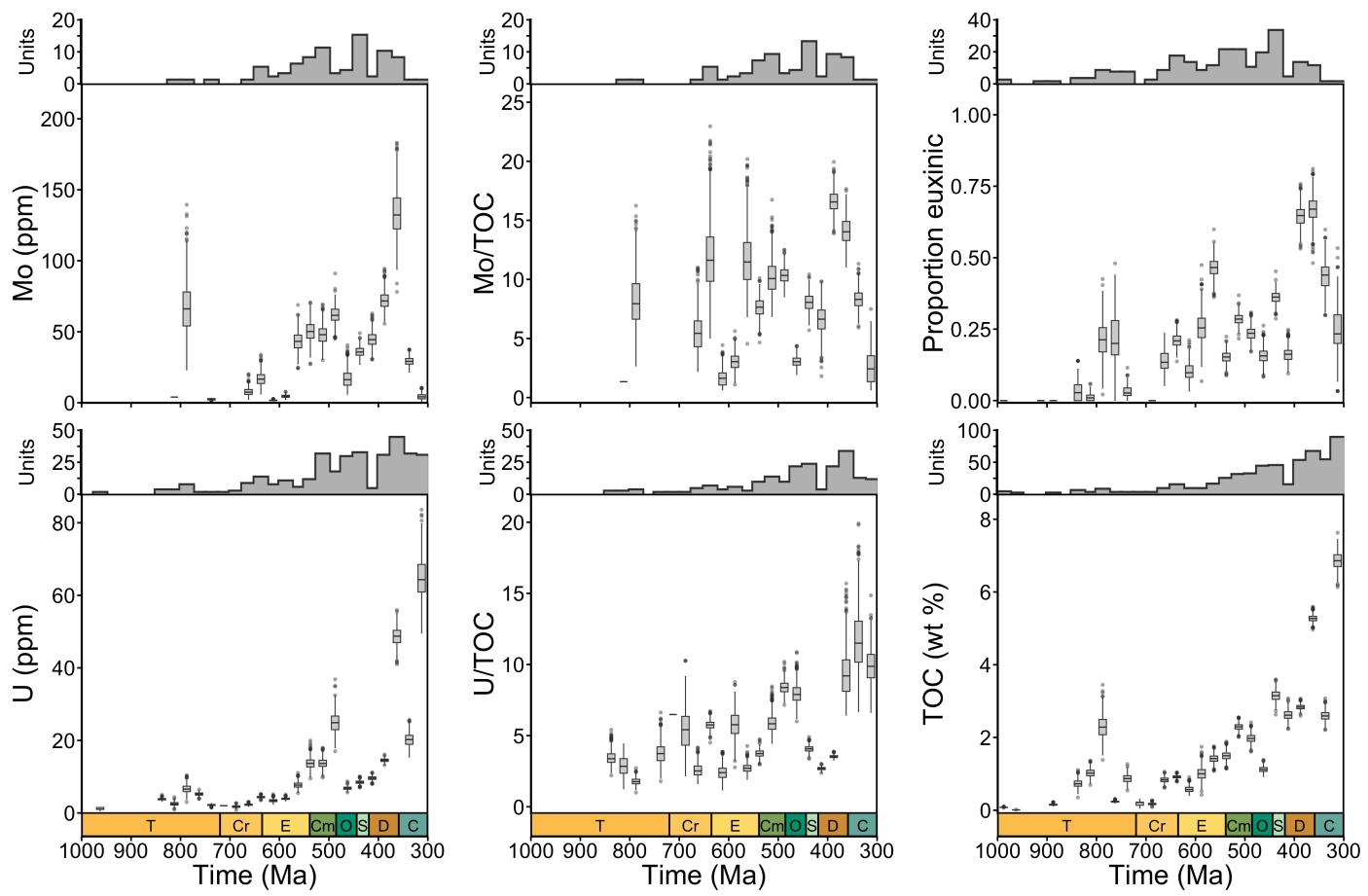
```
## Warning: Removed 4000 rows containing non-finite values (`stat_boxplot()`).
## Removed 4000 rows containing non-finite values (`stat_boxplot()`).
## Removed 4000 rows containing non-finite values (`stat_boxplot()`).
## Removed 4000 rows containing non-finite values (`stat_boxplot()`).
```

```
## Warning: `position_stack()` requires non-overlapping x intervals
## `position_stack()` requires non-overlapping x intervals
## `position_stack()` requires non-overlapping x intervals
```

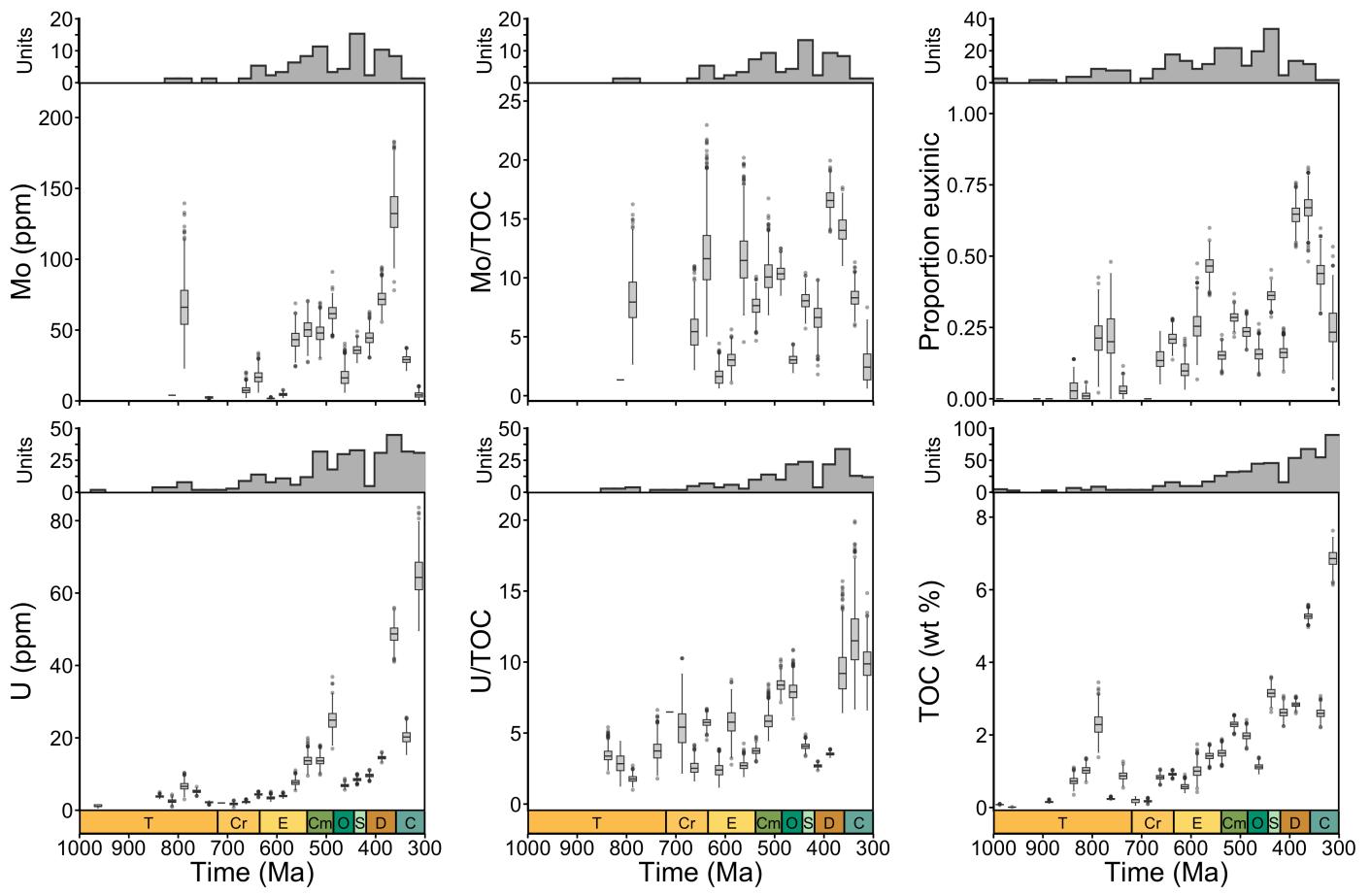
```
## Warning: Removed 4000 rows containing non-finite values (`stat_boxplot()`).
## Removed 4000 rows containing non-finite values (`stat_boxplot()`).
```

```
## Warning: Removed 1000 rows containing non-finite values (`stat_boxplot()`).
## Removed 1000 rows containing non-finite values (`stat_boxplot()`).
```

```
## Warning: Removed 3000 rows containing non-finite values (`stat_boxplot()`).
## Removed 3000 rows containing non-finite values (`stat_boxplot()`).
```



box.fig.composite



```
ggsave(file="Figure 1 Jan 2024 Spatial temporal bootstrap box and whisker plots 20
240208.pdf", box.fig.composite, height=16, width=36)
```

8. Sensitivity analyses of spatial and age scaling factors

Generate bootstrapped means for a loop of age scaling values

```

Molybdenum.sum.age <- as.numeric()
Uranium.sum.age <- as.numeric()
Eux.Fe.sum.age <- as.numeric()
TOC.sum.age <- as.numeric()

for(age.scale.test in c(0.00625, 0.0625, 0.625, 6.25, 62.5)) {

  Molybdenum.mean <- as.numeric()
  Uranium.mean <- as.numeric()
  Eux.Fe.mean <- as.numeric()
  TOC.sum.mean <- as.numeric()

  Molybdenum <- temp.spat.boot(data = Mo.eux.box, var = "Mo..ppm.", reps = 1000, bin_size = bin_size, spatialScale = spatial.scale, ageScale = age.scale.test)

  Molybdenum.mean <- Molybdenum %>%
    group_by(time.bin) %>%
    summarize(mean.boot.mean = mean(boot.mean, na.rm = T))

  Molybdenum.mean$age.scale.test <- age.scale.test

  Uranium <- temp.spat.boot(data = U.anox.box, var = "U..ppm.", reps = 1000, bin_size = bin_size, spatialScale = spatial.scale, ageScale = age.scale.test)

  Uranium.mean <- Uranium %>%
    group_by(time.bin) %>%
    summarize(mean.boot.mean = mean(boot.mean, na.rm = T))

  Uranium.mean$age.scale.test <- age.scale.test

  Eux.Fe <- temp.spat.boot(data = Fepy.anox.box, var = "euxinic.Fe", reps = 1000, bin_size = bin_size, spatialScale = spatial.scale, ageScale = age.scale.test)

  Eux.Fe.mean <- Eux.Fe %>%
    group_by(time.bin) %>%
    summarize(mean.boot.mean = mean(boot.mean, na.rm = T))

  Eux.Fe.mean$age.scale.test <- age.scale.test

  TOC <- temp.spat.boot(data = TOC.all.box, var = "TOC..wt..", reps = 1000, bin_size = bin_size, spatialScale = spatial.scale, ageScale = age.scale.test)

  TOC.mean <- TOC %>%
    group_by(time.bin) %>%
    summarize(mean.boot.mean = mean(boot.mean, na.rm = T))

  TOC.mean$age.scale.test <- age.scale.test

  Molybdenum.sum.age <- rbind(Molybdenum.sum.age, Molybdenum.mean)
  Uranium.sum.age <- rbind(Uranium.sum.age, Uranium.mean)
  Eux.Fe.sum.age <- rbind(Eux.Fe.sum.age, Eux.Fe.mean)
  TOC.sum.age <- rbind(TOC.sum.age, TOC.mean)
}


```

Plot spatial scaling sensitivity analysis

```
Mo.box.composite.age.test <- ggplot(Molybdenum.sum.age, aes(x=time.bin, y=mean.boot.mean))+  
  geom_point(aes(fill = age.scale.test, group = age.scale.test), shape = 21, color = "grey20", size=6, alpha = 0.6)+  
  scale_fill_gradientn(colours=parula(100)[0:90], guide = "colourbar", name = "Age Scale (Myrs)", trans = "log", breaks = c(0.00625, 0.0625, 0.625, 6.25, 62.5), labels = c("0.00625", "0.0625", "0.625", "6.25", "62.5"))+  
  coord_cartesian(xlim=c(start_age+1,end_age-1), ylim=c(-.5,225),expand=FALSE)+  
  theme_minimal()+scale_x_reverse(breaks=c(300,400,500,600,700,800,900,1000))+#breaks=c(360,400,440,480))+  
  ylab("Mo (ppm)")+xlab("Time (Ma)")+  
  theme(plot.margin = ggplot2::margin(0,1,0,1,"cm"),panel.border = element_rect(fill=NA,color="black", size=2,linetype="solid"),  
    axis.ticks = element_line(size=1),  
    axis.line = element_line(lineend = 'square'),  
    axis.title = element_text(size=34),  
    axis.text = element_text( size=26, color="black"),  
    legend.title = element_text(size=24),  
    legend.text = element_text( size=18),  
    axis.title.x = element_blank(),  
    axis.text.x = element_blank(),  
    axis.ticks.length = unit(5, "points"),  
    legend.position="top",  
    legend.title.align = 0.5,  
    panel.grid.major = element_blank(),panel.grid.minor = element_blank())+  
  guides(fill = guide_colorbar(barwidth = 30, title.position="top"))
```

```
U.box.composite.age.test <- ggplot(Uranium.sum.age, aes(x=time.bin, y=mean.boot.mean))+  
  geom_point(aes(fill = age.scale.test, group = age.scale.test), shape = 21, color = "grey20", size=6, alpha = 0.6)+  
  scale_fill_gradientn(colours=parula(100)[0:90], guide = "colourbar", name = "Age Scale (Myrs)", trans = "log", breaks = c(0.00625, 0.0625, 0.625, 6.25, 62.5), labels = c("0.00625", "0.0625", "0.625", "6.25", "62.5"))+  
  coord_geo(xlim=c(start_age+1,end_age-1), ylim=c(-.25,88),expand=FALSE,  
    pos = as.list(rep("bottom", 1)),  
    abbrv=list( T),  
    dat = list(periods.edit),  
    height = list(unit(2, "lines")),  
    size=list(7),  
    bord=list(c("left", "bottom", "right")), lwd=as.list(1))+  
  theme_minimal()+scale_x_reverse(breaks=c(300,400,500,600,700,800,900,1000))+#breaks=c(360,400,440,480))+  
  ylab("U (ppm)")+xlab("Time (Ma)")+  
  theme(plot.margin = ggplot2::margin(0,1,0,1,"cm"),panel.border = element_rect(fill=NA,color="black", size=2,linetype="solid"),  
    axis.ticks = element_line(size=1),  
    axis.line = element_line(lineend = 'square'),  
    axis.title = element_text(size=34),  
    axis.text = element_text( size=26, color="black"),
```

```

legend.title = element_text(size=24),
legend.text = element_text( size=18),
axis.ticks.length = unit(5, "points"),
legend.position="top",
legend.title.align = 0.5,
panel.grid.major = element_blank(),panel.grid.minor = element_blank())+
guides(fill = guide_colorbar(barwidth = 30, title.position="top"))

Fepy.box.composite.age.test <- ggplot(Eux.Fe.sum.age, aes(x=time.bin, y=mean.boot.mean))+

  geom_point(aes(fill = age.scale.test, group = age.scale.test), shape = 21, color = "grey20", size=6, alpha = 0.6)+

    scale_fill_gradientn(colours=parula(100)[0:90], guide = "colourbar", name = "Age Scale (Myrs)", trans = "log", breaks = c(0.00625, 0.0625, 0.625, 6.25, 62.5), labels = c("0.00625", "0.0625", "0.625", "6.25", "62.5"))+

    coord_cartesian(c(start_age+1,end_age-1), ylim=c(-.01,1.11), expand=FALSE)+

    theme_minimal() + scale_x_reverse(breaks=c(300,400,500,600,700,800,900,1000)) + #breaks=c(360,400,440,480))+

    ylab("Proportion euxinic") + xlab("Time (Ma)") +

    theme(plot.margin = ggplot2::margin(0,1,0,1,"cm"), panel.border = element_rect(fill=NA,color="black", size=2,linetype="solid")),

      axis.ticks = element_line(size=1),
      axis.line = element_line(lineend = 'square'),
      axis.title = element_text(size=34),
      axis.text = element_text( size=26, color="black"),
      legend.title = element_text(size=24),
      legend.text = element_text( size=18),
      axis.title.x = element_blank(),
      axis.text.x = element_blank(),
      axis.ticks.length = unit(5, "points"),
      legend.position="top",
      legend.title.align = 0.5,
      panel.grid.major = element_blank(),panel.grid.minor = element_blank())+
guides(fill = guide_colorbar(barwidth = 30, title.position="top"))

TOC.box.composite.age.test <- ggplot(TOC.sum.age, aes(x=time.bin, y=mean.boot.mean))+

  geom_point(aes(fill = age.scale.test, group = age.scale.test), shape = 21, color = "grey20", size=6, alpha = 0.6)+

    scale_fill_gradientn(colours=parula(100)[0:90], guide = "colourbar", name = "Age Scale (Myrs)", trans = "log", breaks = c(0.00625, 0.0625, 0.625, 6.25, 62.5), labels = c("0.00625", "0.0625", "0.625", "6.25", "62.5"))+

    coord_geo(xlim=c(start_age+1,end_age-1), ylim=c(-.1,8.7),expand=FALSE,
              pos = as.list(rep("bottom", 1)),
              abbrv=list( T),
              dat = list(periods.edit),
              height = list(unit(2, "lines")),
              size=list(7),
              bord=list(c("left", "bottom", "right")), lwd=as.list(1))+

    theme_minimal() + scale_x_reverse(breaks=c(300,400,500,600,700,800,900,1000)) +

    ylab("TOC (wt %)") + xlab("Time (Ma)") +

    theme(plot.margin = ggplot2::margin(0,1,1,1,"cm"), panel.border = element_rect(fill=NA,color="black", size=2,linetype="solid")),

```

```
axis.ticks = element_line(size=1),
axis.line = element_line(lineend = 'square'),
axis.title = element_text(size=34),
axis.text = element_text( size=26, color="black"),
legend.title = element_text(size=24),
legend.text = element_text( size=18),
axis.ticks.length = unit(5, "points"),
legend.position="top",
legend.title.align = 0.5,
panel.grid.major = element_blank(), panel.grid.minor = element_blank())+
guides(fill = guide_colorbar(barwidth = 30, title.position="top"))
```

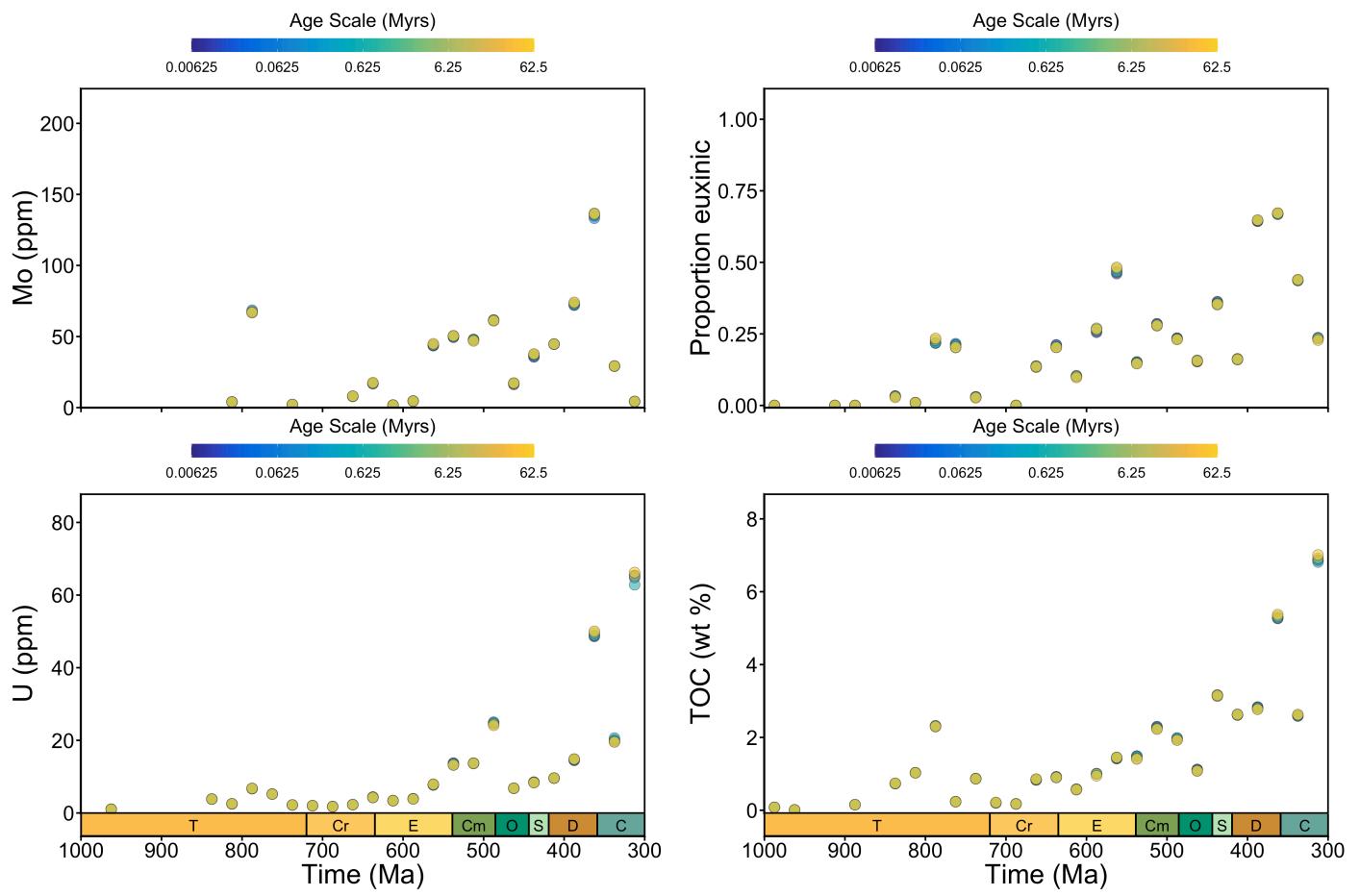
Combine 4 box and whisker plot panels and associated histograms.

```
box.fig.composite.age.test <- ggarrange2(
  Mo.box.composite.age.test,
  Fepy.box.composite.age.test,
  U.box.composite.age.test,
  TOC.box.composite.age.test,
  ncol = 2,
  heights = c(1,1)
)
```

```
## Warning: Removed 15 rows containing missing values (`geom_point()`).
```

```
## Warning: Removed 20 rows containing missing values (`geom_point()`).
## Removed 20 rows containing missing values (`geom_point()`).
```

```
## Warning: Removed 15 rows containing missing values (`geom_point()`).
```



```
ggsave(file="Figure Sx Spatial temporal bootstrap box and whisker plots (age sensitivity) 20240208.pdf", box.fig.composite.age.test, height=16, width=24)
```

Generate bootstrapped means for a loop of age scaling values

```

Molybdenum.sum.space <- as.numeric()
Uranium.sum.space <- as.numeric()
Eux.Fe.sum.space <- as.numeric()
TOC.sum.space <- as.numeric()

for (spatial.scale.test in c(0.005, 0.05, 0.5, 5, 50)*111){

Molybdenum.mean <- as.numeric()
Uranium.mean <- as.numeric()
Eux.Fe.mean <- as.numeric()
TOC.sum.mean <- as.numeric()

Molybdenum <- temp.spat.boot(data = Mo.eux.box, var = "Mo..ppm.", reps = 1000, bin_size = bin_size, spatialScale = spatial.scale.test, ageScale = age.scale)

Molybdenum.mean <- Molybdenum %>%
  group_by(time.bin) %>%
  summarize(mean.boot.mean = mean(boot.mean, na.rm = T))

Molybdenum.mean$spatial.scale.test <- spatial.scale.test

Uranium <- temp.spat.boot(data = U.anox.box, var = "U..ppm.", reps = 1000, bin_size = bin_size, spatialScale = spatial.scale.test, ageScale = age.scale)

Uranium.mean <- Uranium %>%
  group_by(time.bin) %>%
  summarize(mean.boot.mean = mean(boot.mean, na.rm = T))

Uranium.mean$spatial.scale.test <- spatial.scale.test

Eux.Fe <- temp.spat.boot(data = Fepy.anox.box, var = "euxinic.Fe", reps = 1000, bin_size = bin_size, spatialScale = spatial.scale.test, ageScale = age.scale)

Eux.Fe.mean <- Eux.Fe %>%
  group_by(time.bin) %>%
  summarize(mean.boot.mean = mean(boot.mean, na.rm = T))

Eux.Fe.mean$spatial.scale.test <- spatial.scale.test

TOC <- temp.spat.boot(data = TOC.all.box, var = "TOC..wt..", reps = 1000, bin_size = bin_size, spatialScale = spatial.scale.test, ageScale = age.scale)

TOC.mean <- TOC %>%
  group_by(time.bin) %>%
  summarize(mean.boot.mean = mean(boot.mean, na.rm = T))

TOC.mean$spatial.scale.test <- spatial.scale.test

Molybdenum.sum.space <- rbind(Molybdenum.sum.space, Molybdenum.mean)
Uranium.sum.space <- rbind(Uranium.sum.space, Uranium.mean)
Eux.Fe.sum.space <- rbind(Eux.Fe.sum.space, Eux.Fe.mean)
TOC.sum.space <- rbind(TOC.sum.space, TOC.mean)
}

```

Plot spatial scaling sensitivity analysis

```
Mo.box.composite.spatial.test <- ggplot(Molybdenum.sum.space, aes(x=time.bin, y=mean.boot.mean))+  
  geom_point(aes(fill = spatial.scale.test, group = spatial.scale.test), shape = 21,  
             color = "grey20", size=6, alpha = 0.6)+  
  scale_fill_gradientn(colours=parula(100)[0:90], guide = "colourbar", name = "Spatial Scale (km)", trans = "log", breaks = c(0.555,5.55,55.5,555,5550), labels = c("0.555", "5.55", "55.5", "555", "5550"))+  
  coord_cartesian(xlim=c(start_age+1,end_age-1), ylim=c(-.5,225),expand=FALSE)+  
  theme_minimal()+scale_x_reverse(breaks=c(300,400,500,600,700,800,900,1000))+#breaks=c(360,400,440,480))+  
  ylab("Mo (ppm)")+xlab("Time (Ma)")+  
  theme(plot.margin = ggplot2::margin(0,1,0,1,"cm"),panel.border = element_rect(fill=NA,color="black", size=2,linetype="solid"),  
        axis.ticks = element_line(size=1),  
        axis.line = element_line(lineend = 'square'),  
        axis.title = element_text(size=34),  
        axis.text = element_text( size=26, color="black"),  
        legend.title = element_text(size=24),  
        legend.text = element_text( size=18),  
        axis.title.x = element_blank(),  
        axis.text.x = element_blank(),  
        axis.ticks.length = unit(5, "points"),  
        legend.position="top",  
        legend.title.align = 0.5,  
        panel.grid.major = element_blank(),panel.grid.minor = element_blank())+  
  guides(fill = guide_colorbar(barwidth = 30, title.position="top"))
```

```
U.box.composite.spatial.test <- ggplot(Uranium.sum.space, aes(x=time.bin, y=mean.boot.mean))+  
  geom_point(aes(fill = spatial.scale.test, group = spatial.scale.test), shape = 21, color = "grey20", size=6, alpha = 0.6)+  
  scale_fill_gradientn(colours=parula(100)[0:90], guide = "colourbar", name = "Spatial Scale (km)", trans = "log", breaks = c(0.555,5.55,55.5,555,5550), labels = c("0.555", "5.55", "55.5", "555", "5550"))+  
  coord_geo(xlim=c(start_age+1,end_age-1), ylim=c(-.25,88),expand=FALSE,  
            pos = as.list(rep("bottom", 1)),  
            abbrv=list( T),  
            dat = list(periods.edit),  
            height = list(unit(2, "lines")),  
            size=list(7),  
            bord=list(c("left", "bottom", "right")), lwd=as.list(1))+  
  theme_minimal()+scale_x_reverse(breaks=c(300,400,500,600,700,800,900,1000))+#breaks=c(360,400,440,480))+  
  ylab("U (ppm)")+xlab("Time (Ma)")+  
  theme(plot.margin = ggplot2::margin(0,1,0,1,"cm"),panel.border = element_rect(fill=NA,color="black", size=2,linetype="solid"),  
        axis.ticks = element_line(size=1),  
        axis.line = element_line(lineend = 'square'),  
        axis.title = element_text(size=34),  
        axis.text = element_text( size=26, color="black"),
```

```

legend.title = element_text(size=24),
legend.text = element_text( size=18),
axis.ticks.length = unit(5, "points"),
legend.position="top",
legend.title.align = 0.5,
panel.grid.major = element_blank(),panel.grid.minor = element_blank())+
guides(fill = guide_colorbar(barwidth = 30, title.position="top"))

Fepy.box.composite.spatial.test <- ggplot(Eux.Fe.sum.space, aes(x=time.bin, y=mean
.boot.mean))+

  geom_point(aes(fill = spatial.scale.test, group = spatial.scale.test), shape = 2
1, color = "grey20", size=6, alpha = 0.6)+

  scale_fill_gradientn(colours=parula(100)[0:90], guide = "colourbar", name = "Sp
atial Scale (km)", trans = "log", breaks = c(0.555,5.55,55.5,555,5550), labels = c
("0.555", "5.55", "55.5", "555", "5550"))+

  coord_cartesian(c(start_age+1,end_age-1), ylim=c(-.01,1.11), expand=FALSE)+

  theme_minimal() + scale_x_reverse(breaks=c(300,400,500,600,700,800,900,1000)) + #bre
aks=c(360,400,440,480))+

  ylab("Proportion euxinic") + xlab("Time (Ma)") +

  theme(plot.margin = ggplot2::margin(0,1,0,1,"cm"), panel.border = element_rect(fi
ll=NA,color="black", size=2,linetype="solid")),

  axis.ticks = element_line(size=1),
  axis.line = element_line(lineend = 'square'),
  axis.title = element_text(size=34),
  axis.text = element_text( size=26, color="black"),
  legend.title = element_text(size=24),
  legend.text = element_text( size=18),
  axis.title.x = element_blank(),
  axis.text.x = element_blank(),
  axis.ticks.length = unit(5, "points"),
  legend.position="top",
  legend.title.align = 0.5,
  panel.grid.major = element_blank(),panel.grid.minor = element_blank())+
guides(fill = guide_colorbar(barwidth = 30, title.position="top"))

```

```

TOC.box.composite.spatial.test <- ggplot(TOC.sum.space, aes(x=time.bin, y=mean.boo
t.mean))+

  geom_point(aes(fill = spatial.scale.test, group = spatial.scale.test), shape = 2
1, color = "grey20", size=6, alpha = 0.6)+

  scale_fill_gradientn(colours=parula(100)[0:90], guide = "colourbar", name = "Sp
atial Scale (km)", trans = "log", breaks = c(0.555,5.55,55.5,555,5550), labels = c
("0.555", "5.55", "55.5", "555", "5550"))+

  coord_geo(xlim=c(start_age+1,end_age-1), ylim=c(-.1,8.7),expand=FALSE,
  pos = as.list(rep("bottom", 1)),
  abbrv=list( T),
  dat = list(periods.edit),
  height = list(unit(2, "lines")),
  size=list(7),
  bord=list(c("left", "bottom", "right")), lwd=as.list(1))+

  theme_minimal() + scale_x_reverse(breaks=c(300,400,500,600,700,800,900,1000)) +

  ylab("TOC (wt %)") + xlab("Time (Ma)") +

  theme(plot.margin = ggplot2::margin(0,1,1,1,"cm"), panel.border = element_rect(fi
ll=NA,color="black", size=2,linetype="solid")),

```

```
axis.ticks = element_line(size=1),
axis.line = element_line(lineend = 'square'),
axis.title = element_text(size=34),
axis.text = element_text( size=26, color="black"),
legend.title = element_text(size=24),
legend.text = element_text( size=18),
axis.ticks.length = unit(5, "points"),
legend.position="top",
legend.title.align = 0.5,
panel.grid.major = element_blank(), panel.grid.minor = element_blank())+
guides(fill = guide_colorbar(barwidth = 30, title.position="top"))
```

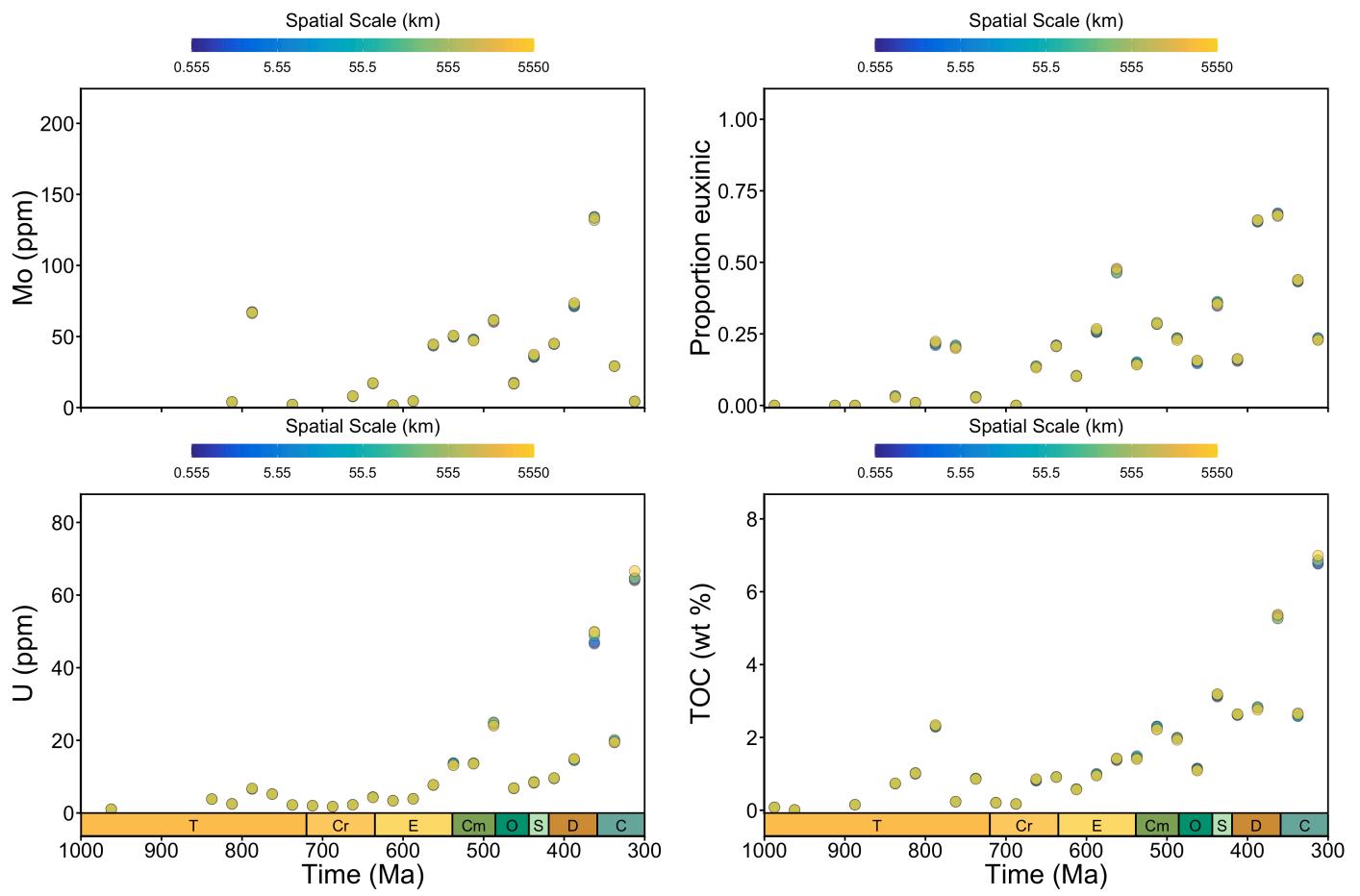
Combine 4 box and whisker plot panels and associated histograms.

```
box.fig.composite.spatial.test <- ggarrange2(
  Mo.box.composite.spatial.test,
  Fepy.box.composite.spatial.test,
  U.box.composite.spatial.test,
  TOC.box.composite.spatial.test,
  ncol = 2,
  heights = c(1,1)
)
```

```
## Warning: Removed 15 rows containing missing values (`geom_point()`).
```

```
## Warning: Removed 20 rows containing missing values (`geom_point()`).
## Removed 20 rows containing missing values (`geom_point()`).
```

```
## Warning: Removed 15 rows containing missing values (`geom_point()`).
```



```
ggsave(file="Figure Sx Spatial temporal bootstrap box and whisker plots (spatial sensitivity) 20240208.pdf", box.fig.composite.spatial.test, height=16, width=24)
```