Doing Basic Analysis

- We now have site information across the Czech Republic, with samples, and with taxon names.
- Let's look at the distributions of taxa across time, simply their presence absence.
- Pick the top 20 taxa (based on the number of times they appear in the records) and look at their distributions in time

In [ ]:

```
taxabyage <- allSamp %>%
  group_by(replacement, "age" = round(age, -2)) %>%
  summarise(n = n())
```

In [ ]:

```
samplesbyage <- allSamp %>%
  group_by("age" = round(age, -2)) %>%
  summarise(samples = length(unique(sampleid)))

taxabyage <- taxabyage %>%
  inner_join(samplesbyage, by = "age") %>%
  mutate(proportion = n / samples)

toptaxa <- taxabyage %>%
  group_by(replacement) %>%
  summarise(n = n()) %>%
  arrange(desc(n)) %>%
  head(n = 10)

groupbyage <- taxabyage %>%
  filter(replacement %in% toptaxa$replacement)
```

In [ ]:

```
ggplot(groupbyage, aes(x = age, y = proportion)) +
  geom_point() +
  geom_smooth(method = 'gam',
              method.args = list(family = 'binomial')) +
  facet_wrap(~replacement) +
  coord_cartesian(xlim = c(0, 20000), ylim = c(0, 1))
```

Stratigraphic Plotting for One Site

In [ ]:

```
# Get a particular site:
onesite <- samples(cz_dl[[1]]) %>%
  inner_join(translation, by = c("variablename" = "variablename")) %>%
  select(!c("variablename", "sites", "samples")) %>%
  group_by(siteid, sitename, replacement,
           sampleid, units, age,
           agetype, depth, datasetid,
           long, lat) %>%
  summarise(value = sum(value))
DT::datatable(head(onesite, n = 20), rownames = FALSE)
```

In [ ]:

```
onesite <- onesite %>%
  filter(units == "NISP") %>%
  group_by(age) %>%
  mutate(pollencount = sum(value, na.rm = TRUE)) %>%
  group_by(replacement) %>%
  mutate(prop = value / pollencount)

topcounts <- onesite %>%
  group_by(replacement) %>%
  summarise(n = n()) %>%
  arrange(desc(n)) %>%
  head(n = 10)
```

In [ ]:

```
widetable <- onesite %>%
  filter(replacement %in% topcounts$replacement) %>%
  select(age, replacement, prop) %>%
  mutate(prop = as.numeric(prop))

counts <- tidyr::pivot_wider(widetable,
                             id_cols = age,
                             names_from = replacement,
                             values_from = prop,
                             values_fill = 0)
```

In [ ]:

```
rioja::strat.plot(counts[,-1], yvar = counts$age,
                  title = cz_dl[[1]]$sitename)
```

Using Spatial-based Data (July max temperature)

In [ ]:

```
modern <- allSamp %>% filter(age < 50)

spatial <- sf::st_as_sf(modern,
                        coords = c("long", "lat"),
                        crs = "+proj=longlat +datum=WGS84")
```

In [ ]:

```
worldTmax <- raster::getData('worldclim', var = 'tmax', res = 10)
worldTmax
```

In [ ]:

```
modern$tmax7 <- raster::extract(worldTmax, spatial)[,7]
head(modern)
```

In [ ]:

```
maxsamp <- modern %>%
  group_by(siteid, sitename) %>%
  dplyr::distinct(tmax7)
head(maxsamp)
```

In [ ]:

```
topten <- allSamp %>%
  dplyr::group_by(replacement) %>%
  dplyr::summarise(n = dplyr::n()) %>%
  dplyr::arrange(desc(n))
head(topten, n=10)
```

In [ ]:

```
pollen_subsamp <- modern %>%
  dplyr::filter(replacement %in% topten$replacement[1:16])
head(pollen_subsamp, n = 5)
```

In [ ]:

```
ggplot() +
  geom_density(data = pollen_subsamp,
               aes(x = round(tmax7 / 10, 0)), col = 2) +
  facet_wrap(~replacement) +
  geom_density(data = maxsamp, aes(x = tmax7 / 10)) +
  xlab("Maximum July Temperature") +
  ylab("Kernel Density")
```