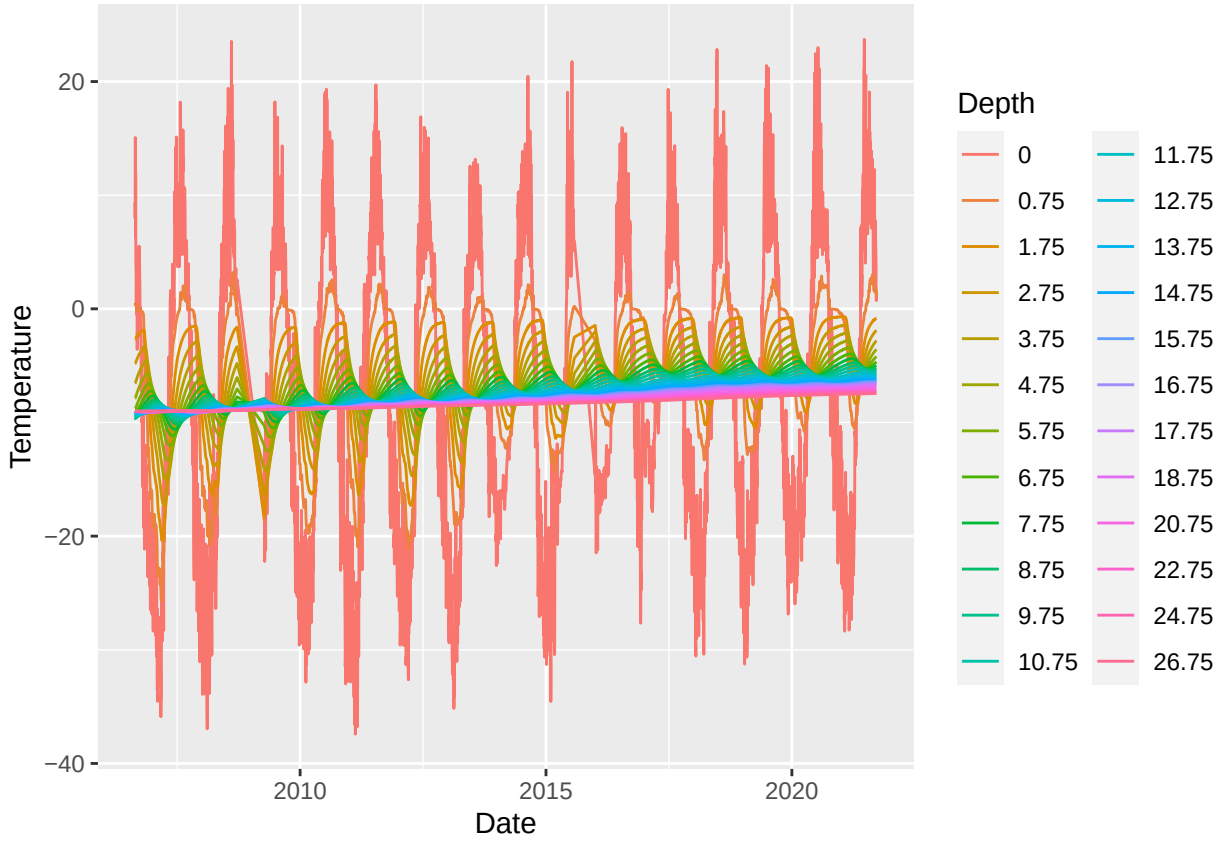
TP - température à Samoylov

```
library(tidyverse)
```

1. Charger la base de données `borehole_Samoylov_byday.csv` et représenter sur un même graphe par profondeur les températures en fonction du temps. Que remarque-t-on ?

```
samoylov = read_csv(file = "borehole_Samoylov_byday.csv", col_names = TRUE) %>%
  rename(Date = Date.Depth)
samoylov_long = samoylov %>%
  gather("Depth", "Temperature", -Date) %>%
  mutate(Depth = as.numeric(substr(Depth, 2, nchar(Depth))))
ggplot(samoylov_long) +
  aes(x = Date, y = Temperature, col = as.factor(Depth)) +
  geom_line() +
  guides(color = guide_legend(title = "Depth"))
```



2. Créer une variable D qui compte le nombre de jours écoulés depuis le 24-08-2006. L'avantage d'introduire une telle variable est de prendre en compte plus facilement les deux périodes sans mesures. La base de données contient aussi 6 valeurs manquantes (`NA`) que nous allons supprimer.

```
origin = as.Date("24-08-2006", format = "%d-%m-%Y")
samoylov = samoylov %>%
```

```
  mutate(day_elapsed = as.numeric(Date) - as.numeric(origin)) %>%
  drop_na()
```

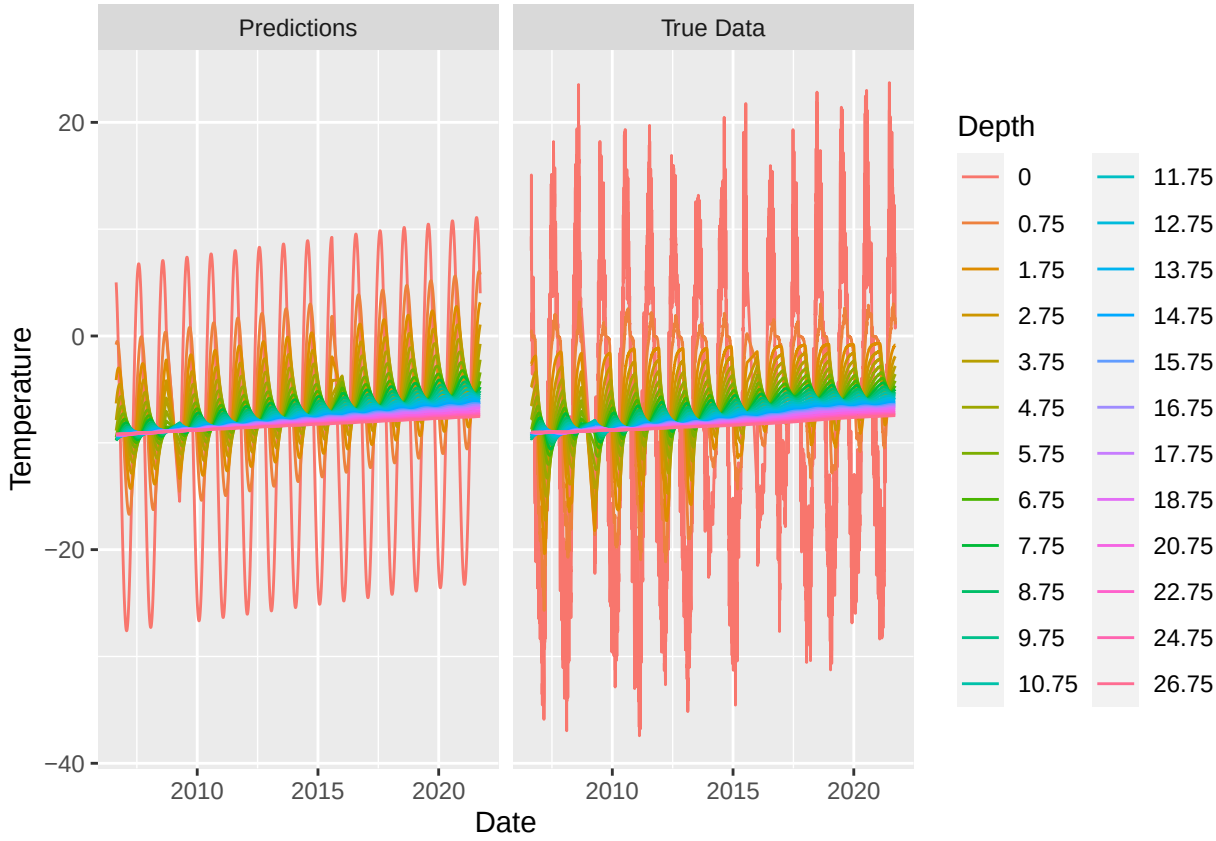
3. Expliquer le choix de ce modèle.

4. Implémenter ce modèle et commenter les résultats. On pourra interpréter, en fonction de la profondeur, la qualité d'ajustement du modèle aux observation et interpréter les coefficients $\widehat{\beta}_{N,1}$ estimés.

```
samoylov = samoylov %>%
  mutate(cos = cos(2 * pi * day_elapsed / 365),
         sin = sin(2 * pi * day_elapsed / 365))
depths = samoylov %>% select(starts_with("X")) %>% names()

models = map_dfr(depths, function(depth) {
  outcome = pull(samoylov[,depth])
  model = lm(outcome ~ day_elapsed + cos + sin, data = samoylov)
  summ = summary(model)
  coefs = summ$coefficients["day_elapsed",] %>% t() %>% as_tibble()
  bind_cols(tibble(Depth = as.numeric(substr(depth, 2, nchar(depth))),
                   Model = list(model)),
            `Adj. R2` = summ$adj.r.squared,
            coefs)
})

models %>%
  select(Depth, `Adj. R2`) %>%
  kable(digits = 3)
```

Depth	Adj. R2
0.00	0.870
0.75	0.816
1.75	0.802
2.75	0.821
3.75	0.839
4.75	0.862
5.75	0.888
6.75	0.913
7.75	0.932
8.75	0.948
9.75	0.958
10.75	0.967
11.75	0.974
12.75	0.979
13.75	0.982
14.75	0.984
15.75	0.985
16.75	0.985
17.75	0.985
18.75	0.984
20.75	0.983
22.75	0.981
24.75	0.979
26.75	0.976

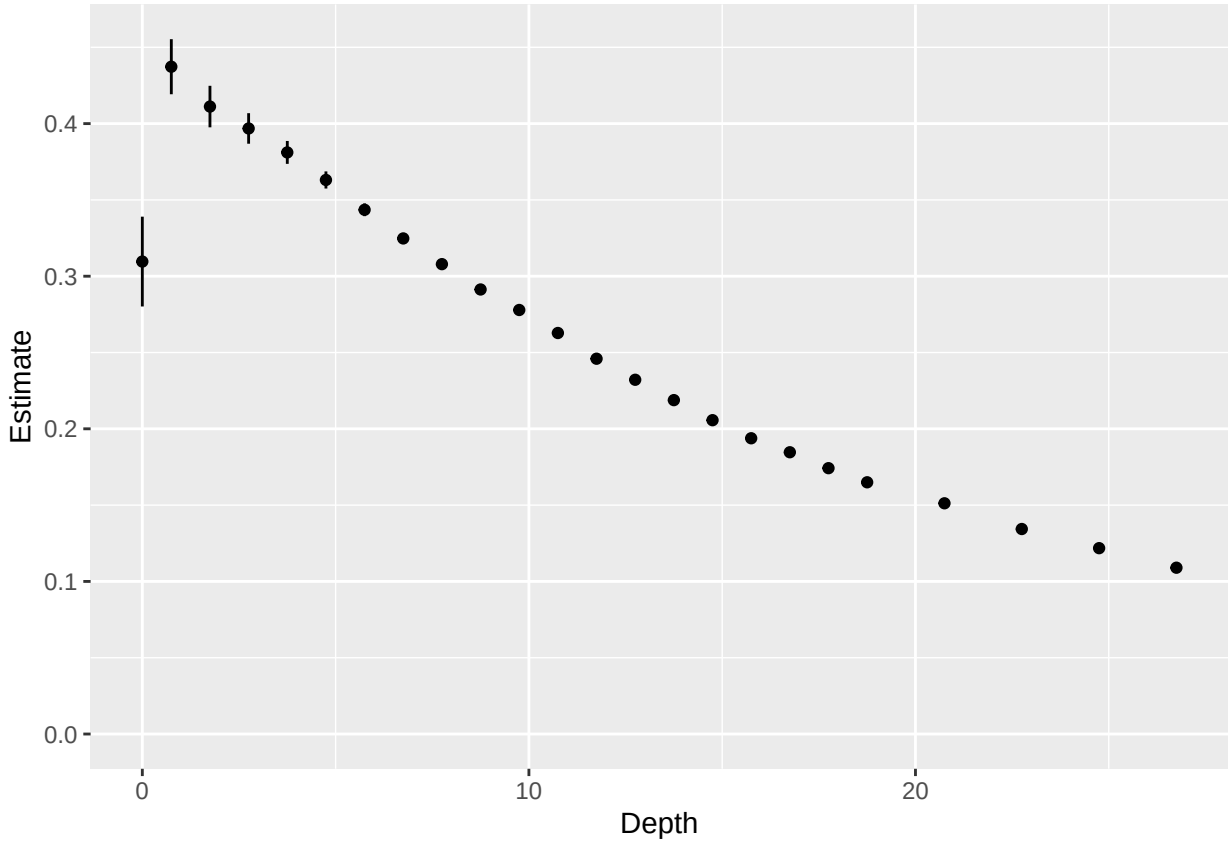
```
predictions = models %>%
  mutate(Predictions = map(Model,
                           ~ tibble(Date = samoylov$Date,
                                    Temperature = predict(.)))) %>%
  select(Depth, Predictions) %>%
  unnest(cols = c("Predictions"))

df = bind_rows(samoylov_long %>% mutate(Origin = "True Data"),
               predictions %>% mutate(Origin = "Predictions"))

ggplot(df) +
  aes(x = Date, y = Temperature, col = as.factor(Depth)) +
  geom_line() +
  facet_wrap(~ Origin, nrow = 1) +
  guides(color = guide_legend(title = "Depth"))
```



5. Donner, par profondeur, l'augmentation annuelle moyenne de température. On donnera à chaque fois un intervalle de confiance et on regardera où se trouve la valeur 0 par rapport à cet intervalle. Mettre en regard ces résultats avec l'augmentation observée sur un siècle du pergélisol au niveau mondial.

```
models_yearly = models %>%
  mutate(Estimate = 365 * Estimate,
         `Std. Error` = 365 * `Std. Error`,
         `Lower CI` = Estimate - 1.96 * `Std. Error`,
         `Upper CI` = Estimate + 1.96 * `Std. Error`)

ggplot(models_yearly) +
```

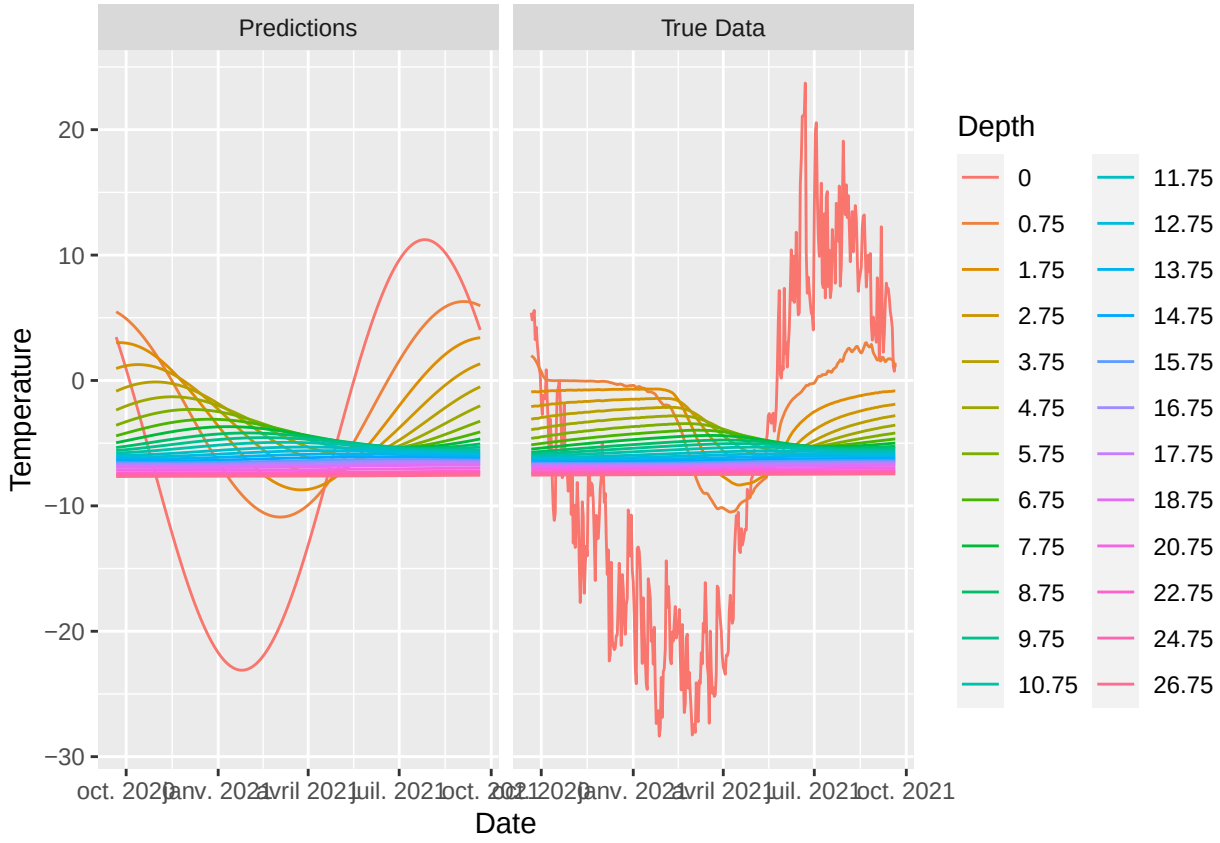
```
  aes(x = Depth, y = Estimate, ymin = `Lower CI`, ymax = `Upper CI`) +
  geom_linerange() +
  geom_point() +
  coord_cartesian(ylim = c(0, NA))
```



6. Etudier la qualité de ce modèle en prévision. Pour cela on pourra par exemple refaire tourner le modèle en laissant de coté une année complète, prédire cette année avec les coefficients estimés et calculer l'erreur de prévision associée.

```
samoylov_truncated = samoylov %>%
  filter(Date <= max(Date) - 365)
samoylov_last_year = samoylov %>%
  filter(Date > max(Date) - 365)
samoylov_last_year_long = samoylov_last_year %>%
  gather("Depth", "Temperature", -Date, -day_elapsed, -cos, -sin) %>%
  mutate(Depth = as.numeric(substr(Depth, 2, nchar(Depth))))

models = map_dfr(depths, function(depth) {
  outcome = pull(samoylov_truncated[,depth])
  model = lm(outcome ~ day_elapsed + cos + sin, data = samoylov_truncated)
  summ = summary(model)
  coefs = summ$coefficients["day_elapsed",] %>% t() %>% as_tibble()
  bind_cols(tibble(Depth = as.numeric(substr(depth, 2, nchar(depth))),
                   Model = list(model)),
            R2 = summ$r.squared,
            coefs)
})
```

```
predictions = models %>%
  mutate(Predictions = map(Model,
    ~ tibble(Date = samoylov_last_year$Date,
             Temperature = predict(., newdata = samoylov_last_year)))) %>%
  select(Depth, Predictions) %>%
  unnest(cols = c("Predictions"))

df = bind_rows(samoylov_last_year_long %>% mutate(Origin = "True Data"),
               predictions %>% mutate(Origin = "Predictions"))

ggplot(df) +
  aes(x = Date, y = Temperature, col = as.factor(Depth)) +
  geom_line() +
  facet_wrap(~ Origin, nrow = 1) +
  guides(color = guide_legend(title = "Depth"))
```




```
df %>%
  select(Date, Depth, Origin, Temperature) %>%
  spread(Origin, Temperature) %>%
  group_by(Depth) %>%
  summarise(MSE = mean((Predictions - `True Data`)^2)) %>%
  kable(digits = 3)
```

Depth	MSE
0.00	21.808
0.75	14.755

Depth	MSE
1.75	8.779
2.75	4.642
3.75	2.562
4.75	1.436
5.75	0.815
6.75	0.503
7.75	0.338
8.75	0.232
9.75	0.172
10.75	0.122
11.75	0.079
12.75	0.053
13.75	0.033
14.75	0.019
15.75	0.009
16.75	0.005
17.75	0.001
18.75	0.000
20.75	0.002
22.75	0.006
24.75	0.010
26.75	0.014