

# Pre Post Analysis

Max Bi

2025-03-02

```
KOH1 = as.Date("2023-04-05")
o1_bp = read_csv("Analysis Data/Obj1BPPrePost.csv")

## New names:
## Rows: 240 Columns: 16
## -- Column specification
## ----- Delimiter: "," chr
## (1): Sex dbl (13): ...1, UniqueIdentifier, sys, dia, KOH, koh.counts, KOH.none,
## KOH.... date (2): BPDate, KOHDate
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `` -> `...1`
```

```
names(o1_bp)[1] = "row_id"
o1_a1c = read_csv("Analysis Data/Obj1A1cPrePost.csv")
```

```
## New names:
## Rows: 394 Columns: 15
## -- Column specification
## ----- Delimiter: "," chr
## (1): Sex dbl (12): ...1, UniqueIdentifier, A1c, KOH, koh.counts, KOH.none,
## KOH.one, ... date (2): A1cDate, KOHDate
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `` -> `...1`
```

```
names(o1_a1c)[1] = "row_id"
o2_bp = read_csv("Analysis Data/Obj2BPPrePost.csv")
```

```
## New names:
## Rows: 9642 Columns: 12
## -- Column specification
## ----- Delimiter: "," chr
## (2): Group, Sex dbl (9): ...1, UniqueIdentifier, Systolic, Diastolic, Marsh,
## age, IncomeLev... date (1): Date
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `` -> `...1`
```

```
names(o2_bp)[1] = "row_id"
o2_a1c = read_csv("Analysis Data/Obj2A1cPrePost.csv")
```

```
## New names:
## Rows: 5118 Columns: 11
## -- Column specification
```

```

## ----- Delimiter: "," chr
## (2): Group, Sex dbl (8): ...1, UniqueIdentifier, A1c, Marsh, age, IncomeLevel,
## BLACERISK, a... date (1): Date
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `` -> `...1`

names(o2_a1c)[1] = "row_id"
DAYS_IN_MONTH = 30.4

# attendees
#   measured before 1st KOH visit: change date to 1st KOH visit
#   measured after 1st KOH visit: keep date
# non-attendees
#   measured before KOH 1: change date to KOH 1
#   measured after KOH 1: DNE
std_times = o1_bp %>%
  group_by(UniqueIdentifier) %>%
  summarize(
    pre_Date = min(BPDate),
    post_Date = max(BPDate),
    std_time = ifelse(BPDate == pre_Date, 0, ifelse(
      KOH == 0,
      as.numeric(post_Date - KOH1),
      ifelse(pre_Date < KOH1, as.numeric(post_Date - KOH1), as.numeric(post_Date - KOHDate))
    ))
  ) %>%
  select(std_time) %>%
  ungroup() %>%
  mutate(
    row_id = as.numeric(o1_bp[["row_id"]]),
    std_months = std_time / DAYS_IN_MONTH
  ) %>%
  select(std_time, std_months, row_id)

## Warning: Returning more (or less) than 1 row per `summarise()` group was deprecated in
## dplyr 1.1.0.
## i Please use `reframe()` instead.
## i When switching from `summarise()` to `reframe()`, remember that `reframe()`
##   always returns an ungrouped data frame and adjust accordingly.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

## `summarise()` has grouped output by 'UniqueIdentifier'. You can override using
## the `.groups` argument.
## Adding missing grouping variables: `UniqueIdentifier`

o1_bp = left_join(o1_bp, std_times, by="row_id")
o1_bp = o1_bp %>% mutate(
  Sex = ifelse(o1_bp$Sex == "M", 1, 0),
  koh_cat = as.factor(case_when(
    KOH.none == 1 ~ "None",
    KOH.one == 1 ~ "One",
    KOH.mult == 1 ~ "Multiple"
  ))
)

```

```

std_times = o1_a1c %>%
  group_by(UniqueIdentifier) %>%
  summarize(
    pre_Date = min(A1cDate),
    post_Date = max(A1cDate),
    std_time = ifelse(A1cDate == pre_Date, 0, ifelse(
      KOH == 0,
      as.numeric(post_Date - KOH1),
      ifelse(pre_Date < KOH1, as.numeric(post_Date - KOH1), as.numeric(post_Date - KOHDate))
    ))
  ) %>%
  select(std_time) %>%
  ungroup() %>%
  mutate(
    row_id = as.numeric(o1_a1c[["row_id"]]),
    std_months = std_time / DAYS_IN_MONTH
  ) %>%
  select(std_time, std_months, row_id)

## Warning: Returning more (or less) than 1 row per `summarise()` group was deprecated in
## dplyr 1.1.0.
## i Please use `reframe()` instead.
## i When switching from `summarise()` to `reframe()`, remember that `reframe()`
## always returns an ungrouped data frame and adjust accordingly.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

## `summarise()` has grouped output by 'UniqueIdentifier'. You can override using
## the `.groups` argument.
## Adding missing grouping variables: `UniqueIdentifier`

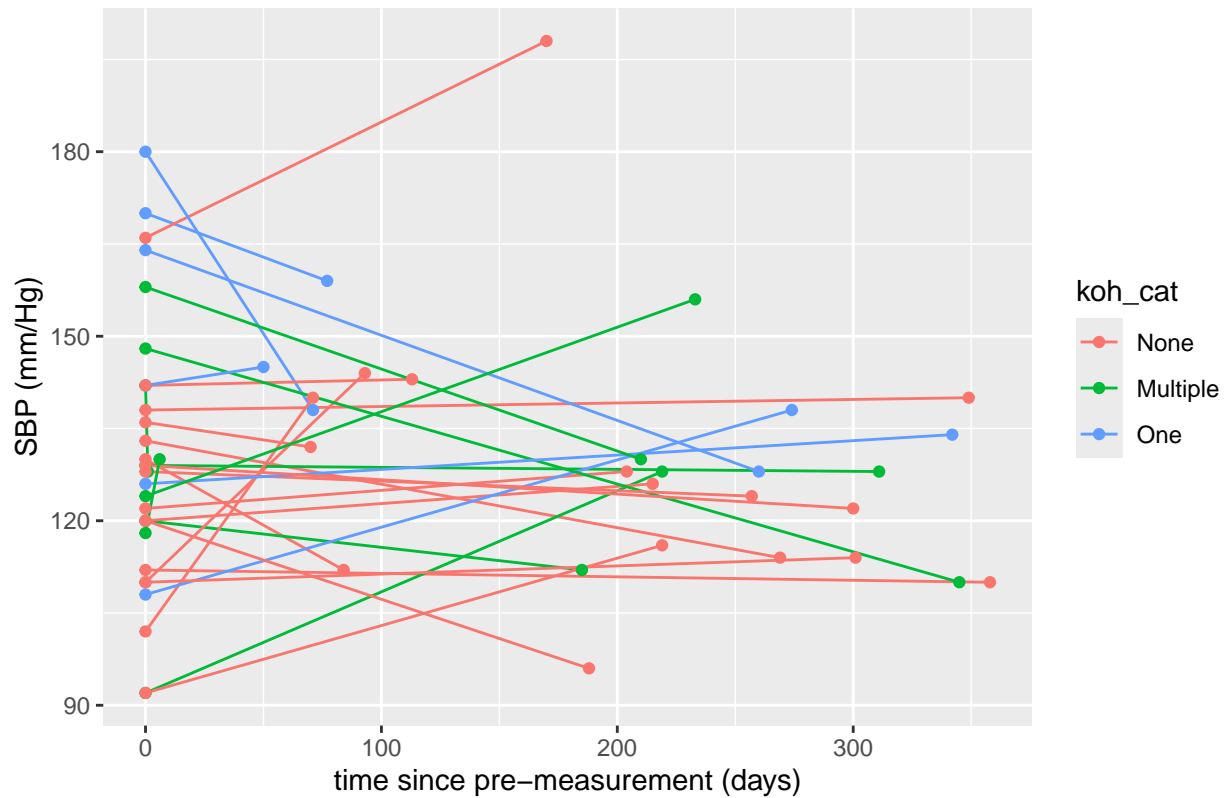
o1_a1c = left_join(o1_a1c, std_times, by="row_id")
o1_a1c = o1_a1c %>% mutate(
  Sex = ifelse(o1_a1c$Sex == "M", 1, 0),
  koh_cat = as.factor(case_when(
    KOH.none == 1 ~ "None",
    KOH.one == 1 ~ "One",
    KOH.mlt == 1 ~ "Multiple"
  ))
)

o1_bp$koh_cat <- relevel(o1_bp$koh_cat, ref = "None")
o1_a1c$koh_cat <- relevel(o1_a1c$koh_cat, ref = "None")

o1_bp_ids = o1_bp$UniqueIdentifier %>% unique() %>% sample(0.5 * 0.25 * nrow(o1_bp))
o1_bp_pdt = o1_bp %>% filter(UniqueIdentifier %in% o1_bp_ids)
ggplot(data=o1_bp_pdt, mapping=aes(x=std_time, y=sys, group=UniqueIdentifier, color=koh_cat)) +
  geom_line() +
  geom_point() +
  labs(
    title="Spaghetti Plot of SBP Against Time",
    x="time since pre-measurement (days)",
    y="SBP (mm/Hg)"
  )

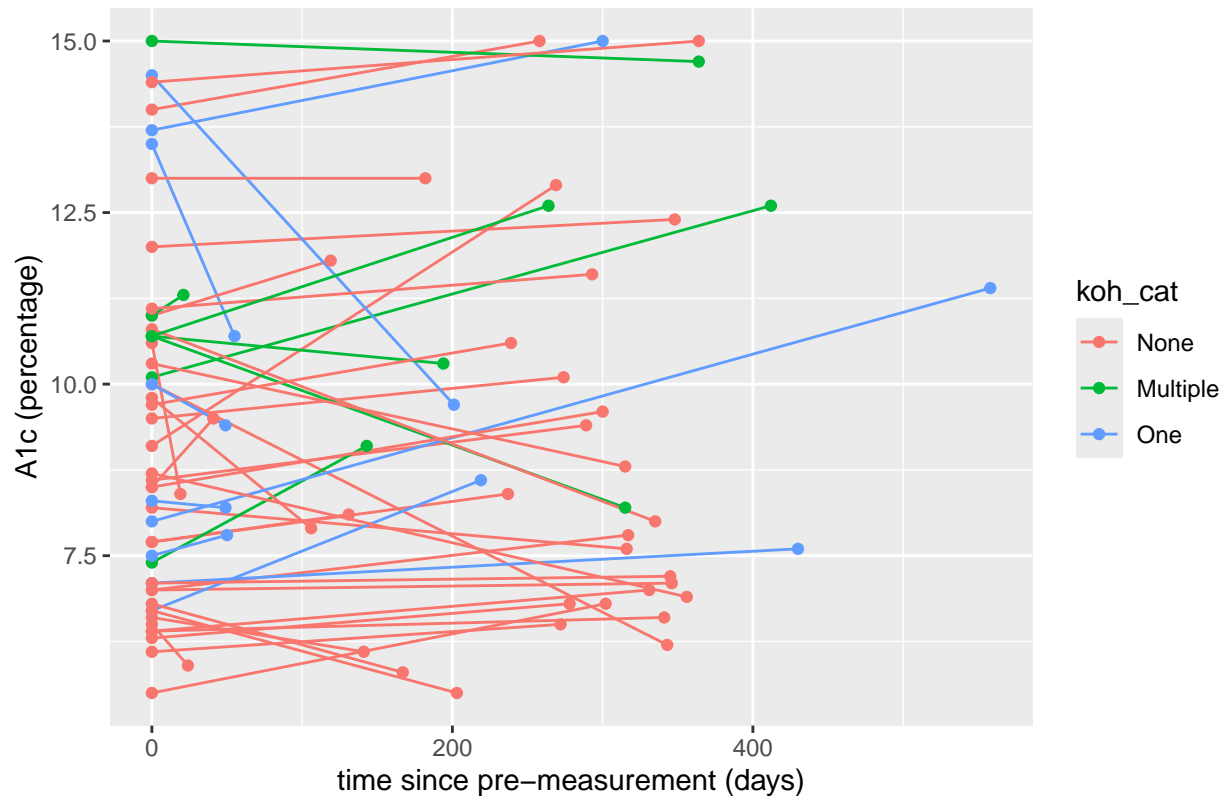
```

Spaghetti Plot of SBP Against Time



```
o1_a1c_ids = o1_a1c$UniqueIdentifier %>% unique() %>% sample(0.5 * 0.25 * nrow(o1_a1c))
o1_a1c_pdt = o1_a1c %>% filter(UniqueIdentifier %in% o1_a1c_ids)
ggplot(data=o1_a1c_pdt, mapping=aes(x=std_time, y=A1c, group=UniqueIdentifier, color=koh_cat)) +
  geom_line() +
  geom_point() +
  labs(
    title="Spaghetti Plot of A1c Against Time",
    x="time since pre-measurement (days)",
    y="A1c (percentage)"
  )
)
```

Spaghetti Plot of A1c Against Time



```
mod_o1_bp = lme(
  fixed = sys ~ std_months * koh_cat + age + Sex + IncomeLevel + BLACERISK + avg.bmi,
  random = ~ std_months | UniqueIdentifier,
  data = o1_bp,
  method = "REML"
)
summary(mod_o1_bp)
```

```
## Linear mixed-effects model fit by REML
## Data: o1_bp
## AIC BIC logLik
## 2116 2167 -1043
##
## Random effects:
## Formula: ~std_months | UniqueIdentifier
## Structure: General positive-definite, Log-Cholesky parametrization
## StdDev Corr
## (Intercept) 1.16e+01 (Intr)
## std_months 9.07e-04 0
## Residual 1.64e+01
##
## Fixed effects: sys ~ std_months * koh_cat + age + Sex + IncomeLevel + BLACERISK + avg.bmi
## Value Std.Error DF t-value p-value
## (Intercept) 100.8 14.72 117 6.84 0.0000
## std_months 0.0 0.28 117 -0.02 0.9871
## koh_catMultiple 1.5 4.68 112 0.31 0.7534
## koh_catOne 1.8 5.68 112 0.31 0.7556
```

```

## age                0.2        0.16 112      1.00 0.3216
## Sex                5.8        3.17 112      1.81 0.0724
## IncomeLevel        0.0        0.03 112     -0.01 0.9954
## BLACERISK         -0.6        0.74 112     -0.86 0.3940
## avg.bmi            0.6        0.30 112      1.99 0.0495
## std_months:koh_catMultiple 0.1        0.61 117      0.11 0.9149
## std_months:koh_catOne    0.4        0.76 117      0.53 0.5955
## Correlation:
##                  (Intr) std_mn kh_ctM kh_ct0 age      Sex      IncmLv
## std_months      -0.006
## koh_catMultiple  0.041  0.237
## koh_catOne       0.040  0.202  0.196
## age             -0.757 -0.042 -0.207 -0.124
## Sex             -0.101 -0.044 -0.008 -0.034  0.036
## IncomeLevel     -0.221  0.014 -0.064  0.118  0.374 -0.176
## BLACERISK       -0.100 -0.002 -0.036  0.032 -0.159 -0.194 -0.060
## avg.bmi         -0.746 -0.063  0.058 -0.044  0.185  0.054 -0.132
## std_months:koh_catMultiple 0.033 -0.452 -0.515 -0.099  0.026  0.028 -0.035
## std_months:koh_catOne    -0.041 -0.369 -0.088 -0.516  0.051  0.065 -0.010
##                  BLACER avg.bm st_:_M
## std_months
## koh_catMultiple
## koh_catOne
## age
## Sex
## IncomeLevel
## BLACERISK
## avg.bmi                0.131
## std_months:koh_catMultiple -0.035 -0.017
## std_months:koh_catOne    -0.045  0.060  0.167
##
## Standardized Within-Group Residuals:
##      Min      Q1      Med      Q3      Max
## -2.4366 -0.5780 -0.0767  0.4984  2.7401
##
## Number of Observations: 240
## Number of Groups: 120
mod_o1_a1c = lme(
  fixed = A1c ~ std_months * koh_cat + age + Sex + IncomeLevel + BLACERISK + avg.bmi,
  random = ~ std_months | UniqueIdentifier,
  data = o1_a1c,
  method = "REML"
)
summary(mod_o1_a1c)

## Linear mixed-effects model fit by REML
##   Data: o1_a1c
##      AIC   BIC logLik
##   1770 1829   -870
##
## Random effects:
##   Formula: ~std_months | UniqueIdentifier
##   Structure: General positive-definite, Log-Cholesky parametrization
##              StdDev Corr

```

```

## (Intercept) 2.235 (Intr)
## std_months 0.153 -0.483
## Residual 1.138
##
## Fixed effects: A1c ~ std_months * koh_cat + age + Sex + IncomeLevel + BLACERISK + avg.bmi
##
## Value Std.Error DF t-value p-value
## (Intercept) 14.97 1.358 194 11.02 0.0000
## std_months -0.05 0.021 194 -2.45 0.0153
## koh_catMultiple -0.38 0.485 189 -0.77 0.4398
## koh_catOne 0.31 0.535 189 0.58 0.5648
## age -0.06 0.015 189 -3.71 0.0003
## Sex 0.00 0.325 189 0.01 0.9956
## IncomeLevel 0.00 0.002 189 0.53 0.5991
## BLACERISK -0.05 0.082 189 -0.62 0.5378
## avg.bmi -0.07 0.029 189 -2.38 0.0183
## std_months:koh_catMultiple 0.04 0.047 194 0.76 0.4487
## std_months:koh_catOne 0.02 0.055 194 0.37 0.7102
## Correlation:
## (Intr) std_mn kh_ctM kh_ctO age Sex IncmLv
## std_months -0.074
## koh_catMultiple 0.075 0.220
## koh_catOne -0.052 0.200 0.199
## age -0.711 -0.012 -0.170 -0.174
## Sex -0.066 -0.002 0.037 -0.001 0.015
## IncomeLevel -0.291 -0.010 -0.046 -0.011 0.314 -0.178
## BLACERISK -0.086 -0.010 -0.001 0.120 -0.183 -0.131 -0.035
## avg.bmi -0.747 0.011 -0.059 0.121 0.138 0.000 0.020
## std_months:koh_catMultiple 0.035 -0.453 -0.483 -0.089 0.000 0.001 0.000
## std_months:koh_catOne 0.040 -0.385 -0.083 -0.434 -0.003 0.013 -0.024
## BLACER avg.bm st_:_M
## std_months
## koh_catMultiple
## koh_catOne
## age
## Sex
## IncomeLevel
## BLACERISK
## avg.bmi 0.078
## std_months:koh_catMultiple 0.012 -0.003
## std_months:koh_catOne 0.005 -0.013 0.174
##
## Standardized Within-Group Residuals:
## Min Q1 Med Q3 Max
## -1.8968 -0.4062 -0.0485 0.3958 2.0901
##
## Number of Observations: 394
## Number of Groups: 197

# measured before KOH 1: change date to KOH 1
# measured after KOH 1: keep date
std_times = o2_bp %>%
  group_by(UniqueIdentifier) %>%
  summarize(
    pre_Date = min(Date),

```

```

    post_Date = max(Date),
    std_time = ifelse(
      Date == pre_Date,
      0,
      # ifelse(pre_Date < KOH1, as.numeric(post_Date - KOH1), as.numeric(post_Date - pre_Date))
      as.numeric(post_Date - pre_Date)
    )
  ) %>%
  select(std_time) %>%
  ungroup() %>%
  mutate(
    row_id = as.numeric(o2_bp[["row_id"]]),
    std_months = std_time / DAYS_IN_MONTH
  ) %>%
  select(std_time, std_months, row_id)

```

```

## Warning: Returning more (or less) than 1 row per `summarise()` group was deprecated in
## dplyr 1.1.0.
## i Please use `reframe()` instead.
## i When switching from `summarise()` to `reframe()`, remember that `reframe()`
## always returns an ungrouped data frame and adjust accordingly.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

```

```

## `summarise()` has grouped output by 'UniqueIdentifier'. You can override using
## the `.groups` argument.
## Adding missing grouping variables: `UniqueIdentifier`

```

```

o2_bp = left_join(o2_bp, std_times, by="row_id")
o2_bp = o2_bp %>% mutate(Sex = ifelse(o2_bp$Sex == "M", 1, 0))

std_times = o2_a1c %>%
  group_by(UniqueIdentifier) %>%
  summarize(
    pre_Date = min(Date),
    post_Date = max(Date),
    std_time = ifelse(
      Date == pre_Date,
      0,
      # ifelse(pre_Date < KOH1, as.numeric(post_Date - KOH1), as.numeric(post_Date - pre_Date))
      as.numeric(post_Date - KOH1)
    )
  ) %>%
  select(std_time) %>%
  ungroup() %>%
  mutate(
    row_id = as.numeric(o2_a1c[["row_id"]]),
    std_months = std_time / DAYS_IN_MONTH
  ) %>%
  select(std_time, std_months, row_id)

```

```

## Warning: Returning more (or less) than 1 row per `summarise()` group was deprecated in
## dplyr 1.1.0.
## i Please use `reframe()` instead.
## i When switching from `summarise()` to `reframe()`, remember that `reframe()`

```



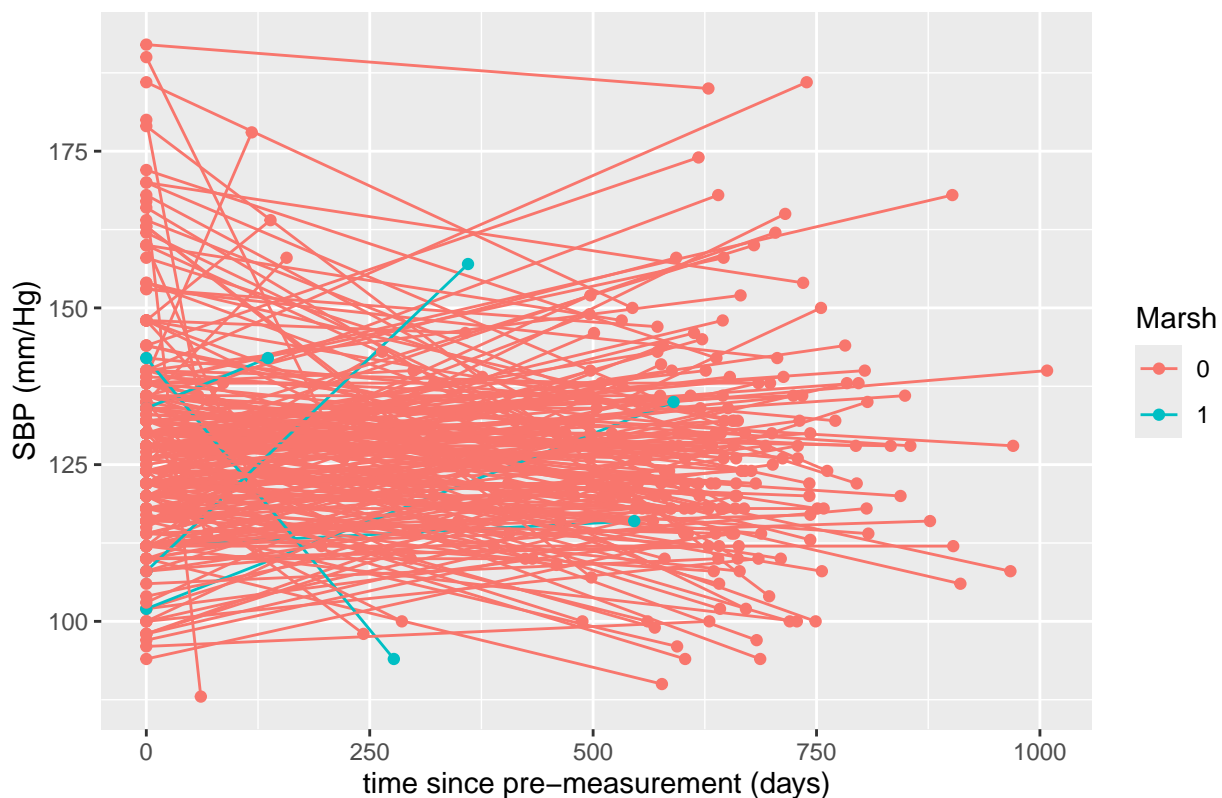
```
## always returns an ungrouped data frame and adjust accordingly.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

## `summarise()` has grouped output by 'UniqueIdentifier'. You can override using
## the `.groups` argument.
## Adding missing grouping variables: `UniqueIdentifier`

o2_a1c = left_join(o2_a1c, std_times, by="row_id")
o2_a1c = o2_a1c %>% mutate(Sex = ifelse(o2_a1c$Sex == "M", 1, 0))

o2_bp_ids = o2_bp$UniqueIdentifier %>% unique() %>% sample(0.5 * 0.05 * nrow(o2_bp))
o2_bp_pdt = o2_bp %>% filter(UniqueIdentifier %in% o2_bp_ids) %>% mutate(Marsh = as.factor(Marsh))
ggplot(data=o2_bp_pdt, mapping=aes(x=std_time, y=Systolic, group=UniqueIdentifier, color=Marsh)) +
  geom_line() +
  geom_point() +
  labs(
    title="Spaghetti Plot of SBP Against Time",
    x="time since pre-measurement (days)",
    y="SBP (mm/Hg)"
  )
)
```

Spaghetti Plot of SBP Against Time



```
o2_a1c_ids = o2_a1c$UniqueIdentifier %>% unique() %>% sample(0.5 * 0.05 * nrow(o2_a1c))
o2_a1c_pdt = o2_a1c %>% filter(UniqueIdentifier %in% o2_a1c_ids) %>% mutate(Marsh = as.factor(Marsh))
ggplot(data=o2_a1c_pdt, mapping=aes(x=std_time, y=A1c, group=UniqueIdentifier, color=Marsh)) +
  geom_line() +
  geom_point() +
  labs(
    title="Spaghetti Plot of A1c Against Time",

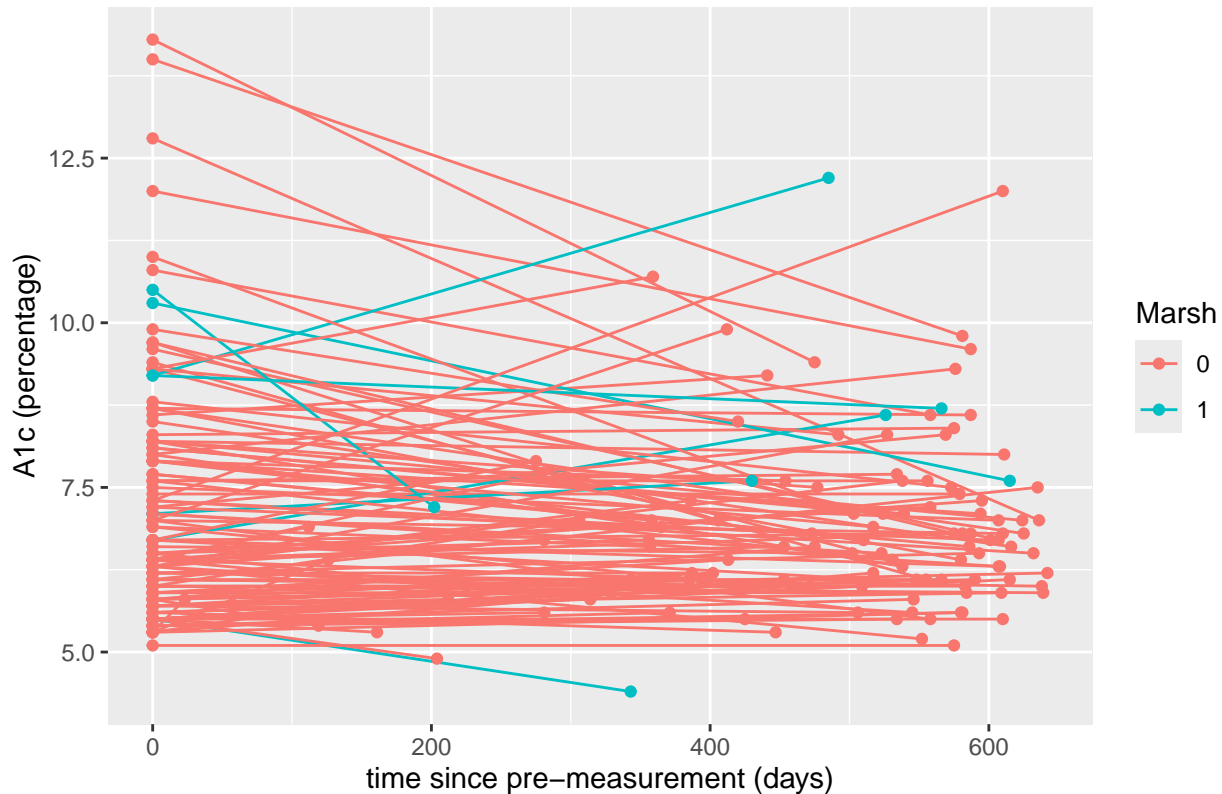
```

```

x="time since pre-measurement (days)",
y="A1c (percentage)"
)

```

Spaghetti Plot of A1c Against Time



```

mod_o2_bp = lme(
  fixed = Systolic ~ std_months * Marsh + age + Sex + IncomeLevel + avg.bmi,
  random = ~ std_months | UniqueIdentifier,
  data = o2_bp,
  method = "REML"
)
summary(mod_o2_bp)

```

```

## Linear mixed-effects model fit by REML
##   Data: o2_bp
##      AIC   BIC logLik
## 80867 80953 -40422
##
## Random effects:
## Formula: ~std_months | UniqueIdentifier
## Structure: General positive-definite, Log-Cholesky parametrization
##              StdDev Corr
## (Intercept)  9.33  (Intr)
## std_months   0.21 -0.293
## Residual     13.48
##
## Fixed effects: Systolic ~ std_months * Marsh + age + Sex + IncomeLevel + avg.bmi
##              Value Std.Error   DF t-value p-value

```

```

## (Intercept)      123.6      1.351 4819      91.5  0.0000
## std_months      -0.1      0.014 4819     -5.5  0.0000
## Marsh           0.3      1.445 4815      0.2  0.8520
## age             0.0      0.015 4815      2.6  0.0097
## Sex             2.2      0.382 4815      5.7  0.0000
## IncomeLevel     0.0      0.000 4815     -0.5  0.6107
## avg.bmi         0.1      0.023 4815      3.8  0.0002
## std_months:Marsh -0.1      0.087 4819     -1.0  0.3214
## Correlation:
##              (Intr) std_mn Marsh  age    Sex    IncmLv avg.bm
## std_months    -0.088
## Marsh         -0.047  0.094
## age           -0.805 -0.011 -0.006
## Sex           -0.280  0.002  0.014  0.126
## IncomeLevel   -0.040 -0.002  0.030  0.002 -0.036
## avg.bmi       -0.743 -0.010  0.035  0.264  0.119 -0.016
## std_months:Marsh 0.013 -0.158 -0.573  0.000  0.004  0.000  0.003
##
## Standardized Within-Group Residuals:
##      Min      Q1      Med      Q3      Max
## -3.4920 -0.5442 -0.0438  0.4580  5.0072
##
## Number of Observations: 9642
## Number of Groups: 4821

mod_o2_a1c = lme(
  fixed = A1c ~ std_months * Marsh + age + Sex + IncomeLevel + avg.bmi,
  random = ~ std_months | UniqueIdentifier,
  data = o2_a1c,
  method = "REML"
)
summary(mod_o2_a1c)

## Linear mixed-effects model fit by REML
##   Data: o2_a1c
##      AIC   BIC logLik
##  19638 19716  -9807
##
## Random effects:
## Formula: ~std_months | UniqueIdentifier
## Structure: General positive-definite, Log-Cholesky parametrization
##              StdDev Corr
## (Intercept) 1.5986 (Intr)
## std_months  0.0549 -0.456
## Residual    0.9979
##
## Fixed effects: A1c ~ std_months * Marsh + age + Sex + IncomeLevel + avg.bmi
##              Value Std.Error   DF t-value p-value
## (Intercept)   8.35   0.2355 2557   35.5  0.0000
## std_months    0.00   0.0022 2557   -0.7  0.4538
## Marsh         2.29   0.1366 2553   16.8  0.0000
## age          -0.01   0.0026 2553   -4.5  0.0000
## Sex           0.26   0.0654 2553    4.0  0.0001
## IncomeLevel   0.00   0.0000 2553   -0.6  0.5674
## avg.bmi      -0.02   0.0038 2553   -5.3  0.0000

```

```

## std_months:Marsh -0.01    0.0076 2557    -1.6  0.1175
## Correlation:
##          (Intr) std_mn Marsh  age    Sex    IncmLv avg.bm
## std_months    -0.072
## Marsh          -0.186  0.140
## age            -0.796 -0.010  0.059
## Sex            -0.265  0.005  0.068  0.059
## IncomeLevel    -0.028 -0.001  0.029  0.018 -0.013
## avg.bmi        -0.765 -0.006  0.163  0.275  0.175 -0.022
## std_months:Marsh 0.021 -0.287 -0.491  0.001  0.001  0.000  0.002
##
## Standardized Within-Group Residuals:
##      Min      Q1      Med      Q3      Max
## -3.211 -0.330 -0.126  0.193  4.934
##
## Number of Observations: 5118
## Number of Groups: 2559

```

## Tables of Regression Results

```
#####
# format for regression results from models
#####

coef_names = c("std_months:koh_catOne", "std_months:koh_catMultiple", "koh_catOne", "koh_catMultiple",
col_names = c("est.", "lower", "upper")

# primary objective
obj1_res = data.frame(
  rbind(
    cbind(intervals(mod_o1_a1c)$fixed[coef_names, col_names], summary(mod_o1_a1c)$tTable[coef_names, "p
    cbind(intervals(mod_o1_bp, which="fixed")$fixed[coef_names, col_names], summary(mod_o1_bp)$tTable[c
  )
)

# obj1_res = data.frame(
#   est = c(summary(mod_o1_a1c)$tTable[coef_names, "Value"], summary(mod_o1_bp)$tTable[coef_names, "Val
#   CI = intervals(mod_o1_a1c)$fixed[coef_names, col_names]
# )

rownames(obj1_res) = c("months:KOHonce", "months:KOHmultiple", "KOHonce", "KOHmultiple", "months", "mon
colnames(obj1_res)[4] = "p-value"

kable(obj1_res) %>% pack_rows("a1c_pre_post", 1, 5) %>% pack_rows("sbp_pre_post", 6, 10)
```

	est.	lower	upper	p-value
<b>a1c_pre_post</b>				
months:KOHonce	0.021	-0.088	0.129	0.710
months:KOHmultiple	0.036	-0.057	0.128	0.449
KOHonce	0.309	-0.747	1.364	0.565
KOHmultiple	-0.375	-1.331	0.581	0.440
months	-0.052	-0.094	-0.010	0.015
<b>sbp_pre_post</b>				
months:KOHonce	0.404	-1.098	1.906	0.596
months:KOHmultiple	0.065	-1.140	1.271	0.915
KOHonce	1.771	-9.474	13.016	0.756
KOHmultiple	1.474	-7.802	10.751	0.753
months	-0.004	-0.551	0.542	0.987

```
# secondary objective
coef_names = c("std_months:Marsh", "Marsh", "std_months")
col_names = c("est.", "lower", "upper")

obj1_res = data.frame(
  rbind(
    cbind(intervals(mod_o2_a1c, which="fixed")$fixed[coef_names, col_names], summary(mod_o2_a1c)$tTable
    cbind(intervals(mod_o2_bp, which="fixed")$fixed[coef_names, col_names], summary(mod_o2_bp)$tTable[c
  )
)
```

```

# obj1_res = data.frame(
#   est = c(summary(mod_o1_a1c)$tTable[coef_names, "Value"], summary(mod_o1_bp)$tTable[coef_names, "Val
#   CI = intervals(mod_o1_a1c)$fixed[coef_names, col_names]
# )

rownames(obj1_res) = c("months:Marsh","Marsh", "months", "months:Marsh ", "Marsh ", "months ")
colnames(obj1_res)[4] = "p-value"

kable(obj1_res) %>% pack_rows("a1c_pre_post", 1, 3) %>% pack_rows("sbp_pre_post", 4, 6)

```

	est.	lower	upper	p-value
<b>a1c_pre_post</b>				
months:Marsh	-0.012	-0.027	0.003	0.118
Marsh	2.292	2.025	2.560	0.000
months	-0.002	-0.006	0.003	0.454
<b>sbp_pre_post</b>				
months:Marsh	-0.087	-0.258	0.085	0.321
Marsh	0.270	-2.564	3.103	0.852
months	-0.076	-0.103	-0.049	0.000

```

# table for undadjusted baseline results
o1_a1c_bltb = o1_a1c %>%
  group_by(UniqueIdentifier) %>%
  summarize(
    pre_Date = min(A1cDate),
    A1c = ifelse(A1cDate == pre_Date, A1c, NA),
    koh_cat = koh_cat
  ) %>%
  drop_na()

```

```

## Warning: Returning more (or less) than 1 row per `summarise()` group was deprecated in
## dplyr 1.1.0.
## i Please use `reframe()` instead.
## i When switching from `summarise()` to `reframe()`, remember that `reframe()`
## always returns an ungrouped data frame and adjust accordingly.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

```

```

## `summarise()` has grouped output by 'UniqueIdentifier'. You can override using
## the `.groups` argument.

```

```

o1_bp_bltb = o1_bp %>%
  group_by(UniqueIdentifier) %>%
  summarize(
    pre_Date = min(BPDate),
    sys = ifelse(BPDate == pre_Date, sys, NA),
    koh_cat = koh_cat
  ) %>%
  drop_na()

```

```

## Warning: Returning more (or less) than 1 row per `summarise()` group was deprecated in
## dplyr 1.1.0.
## i Please use `reframe()` instead.

```

```
## i When switching from `summarise()` to `reframe()`, remember that `reframe()`
## always returns an ungrouped data frame and adjust accordingly.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## `summarise()` has grouped output by 'UniqueIdentifier'. You can override using
## the `.groups` argument.
```

```
o1_bltb = data.frame(
  KOHnever = c(mean((o1_a1c_bltb %>% filter(koh_cat == "None"))$A1c), mean((o1_bp_bltb %>% filter(koh_cat == "None"))$SBP)),
  KOHonce = c(mean((o1_a1c_bltb %>% filter(koh_cat == "One"))$A1c), mean((o1_bp_bltb %>% filter(koh_cat == "One"))$SBP)),
  KOHmultiple = c(mean((o1_a1c_bltb %>% filter(koh_cat == "Multiple"))$A1c), mean((o1_bp_bltb %>% filter(koh_cat == "Multiple"))$SBP))
)
```

```
rownames(o1_bltb) = c("A1c", "SBP")
```

```
o1_bltb
```

```
##      KOHnever KOHonce KOHmultiple
## A1c      9.73    10.1      8.97
## SBP    128.47   130.0    131.26
```

## Code Appendix

```
knitr::opts_chunk$set(echo = TRUE)
library(tidyverse)
library(nlme)
library(kableExtra)
# source("primary_obj_data.R")
# source("secondary_obj_data.R")
options(digits=3)
KOH1 = as.Date("2023-04-05")
o1_bp = read_csv("Analysis Data/Obj1BPPrePost.csv")
names(o1_bp)[1] = "row_id"
o1_a1c = read_csv("Analysis Data/Obj1A1cPrePost.csv")
names(o1_a1c)[1] = "row_id"
o2_bp = read_csv("Analysis Data/Obj2BPPrePost.csv")
names(o2_bp)[1] = "row_id"
o2_a1c = read_csv("Analysis Data/Obj2A1cPrePost.csv")
names(o2_a1c)[1] = "row_id"
DAYS_IN_MONTH = 30.4
# attendees
#   measured before 1st KOH visit: change date to 1st KOH visit
#   measured after 1st KOH visit: keep date
# non-attendees
#   measured before KOH 1: change date to KOH 1
#   measured after KOH 1: DNE
std_times = o1_bp %>%
  group_by(UniqueIdentifier) %>%
  summarize(
    pre_Date = min(BPDate),
    post_Date = max(BPDate),
    std_time = ifelse(BPDate == pre_Date, 0, ifelse(
      KOH == 0,
      as.numeric(post_Date - KOH1),
      ifelse(pre_Date < KOH1, as.numeric(post_Date - KOH1), as.numeric(post_Date - KOHDate))
    ))
  ) %>%
  select(std_time) %>%
  ungroup() %>%
  mutate(
    row_id = as.numeric(o1_bp[["row_id"]]),
    std_months = std_time / DAYS_IN_MONTH
  ) %>%
  select(std_time, std_months, row_id)

o1_bp = left_join(o1_bp, std_times, by="row_id")
o1_bp = o1_bp %>% mutate(
  Sex = ifelse(o1_bp$Sex == "M", 1, 0),
  koh_cat = as.factor(case_when(
    KOH.none == 1 ~ "None",
    KOH.one == 1 ~ "One",
    KOH.mult == 1 ~ "Multiple"
  ))
)
```



```

std_times = o1_a1c %>%
  group_by(UniqueIdentifier) %>%
  summarize(
    pre_Date = min(A1cDate),
    post_Date = max(A1cDate),
    std_time = ifelse(A1cDate == pre_Date, 0, ifelse(
      KOH == 0,
      as.numeric(post_Date - KOH1),
      ifelse(pre_Date < KOH1, as.numeric(post_Date - KOH1), as.numeric(post_Date - KOHDate))
    ))
  ) %>%
  select(std_time) %>%
  ungroup() %>%
  mutate(
    row_id = as.numeric(o1_a1c[["row_id"]]),
    std_months = std_time / DAYS_IN_MONTH
  ) %>%
  select(std_time, std_months, row_id)

o1_a1c = left_join(o1_a1c, std_times, by="row_id")
o1_a1c = o1_a1c %>% mutate(
  Sex = ifelse(o1_a1c$Sex == "M", 1, 0),
  koh_cat = as.factor(case_when(
    KOH.none == 1 ~ "None",
    KOH.one == 1 ~ "One",
    KOH.mult == 1 ~ "Multiple"
  ))
)

o1_bp$koh_cat <- relevel(o1_bp$koh_cat, ref = "None")
o1_a1c$koh_cat <- relevel(o1_a1c$koh_cat, ref = "None")
o1_bp_ids = o1_bp$UniqueIdentifier %>% unique() %>% sample(0.5 * 0.25 * nrow(o1_bp))
o1_bp_pdt = o1_bp %>% filter(UniqueIdentifier %in% o1_bp_ids)
ggplot(data=o1_bp_pdt, mapping=aes(x=std_time, y=sys, group=UniqueIdentifier, color=koh_cat)) +
  geom_line() +
  geom_point() +
  labs(
    title="Spaghetti Plot of SBP Against Time",
    x="time since pre-measurement (days)",
    y="SBP (mm/Hg)"
  )

o1_a1c_ids = o1_a1c$UniqueIdentifier %>% unique() %>% sample(0.5 * 0.25 * nrow(o1_a1c))
o1_a1c_pdt = o1_a1c %>% filter(UniqueIdentifier %in% o1_a1c_ids)
ggplot(data=o1_a1c_pdt, mapping=aes(x=std_time, y=A1c, group=UniqueIdentifier, color=koh_cat)) +
  geom_line() +
  geom_point() +
  labs(
    title="Spaghetti Plot of A1c Against Time",
    x="time since pre-measurement (days)",
    y="A1c (percentage)"
  )
mod_o1_bp = lme(

```

```

    fixed = sys ~ std_months * koh_cat + age + Sex + IncomeLevel + BLACERISK + avg.bmi,
    random = ~ std_months | UniqueIdentifier,
    data = o1_bp,
    method = "REML"
)
summary(mod_o1_bp)

mod_o1_a1c = lme(
  fixed = A1c ~ std_months * koh_cat + age + Sex + IncomeLevel + BLACERISK + avg.bmi,
  random = ~ std_months | UniqueIdentifier,
  data = o1_a1c,
  method = "REML"
)
summary(mod_o1_a1c)
# measured before KOH 1: change date to KOH 1
# measured after KOH 1: keep date
std_times = o2_bp %>%
  group_by(UniqueIdentifier) %>%
  summarize(
    pre_Date = min(Date),
    post_Date = max(Date),
    std_time = ifelse(
      Date == pre_Date,
      0,
      # ifelse(pre_Date < KOH1, as.numeric(post_Date - KOH1), as.numeric(post_Date - pre_Date))
      as.numeric(post_Date - pre_Date)
    )
  ) %>%
  select(std_time) %>%
  ungroup() %>%
  mutate(
    row_id = as.numeric(o2_bp[["row_id"]]),
    std_months = std_time / DAYS_IN_MONTH
  ) %>%
  select(std_time, std_months, row_id)

o2_bp = left_join(o2_bp, std_times, by="row_id")
o2_bp = o2_bp %>% mutate(Sex = ifelse(o2_bp$Sex == "M", 1, 0))

std_times = o2_a1c %>%
  group_by(UniqueIdentifier) %>%
  summarize(
    pre_Date = min(Date),
    post_Date = max(Date),
    std_time = ifelse(
      Date == pre_Date,
      0,
      # ifelse(pre_Date < KOH1, as.numeric(post_Date - KOH1), as.numeric(post_Date - pre_Date))
      as.numeric(post_Date - KOH1)
    )
  ) %>%
  select(std_time) %>%
  ungroup() %>%

```

```

mutate(
  row_id = as.numeric(o2_a1c[["row_id"]]),
  std_months = std_time / DAYS_IN_MONTH
) %>%
select(std_time, std_months, row_id)

o2_a1c = left_join(o2_a1c, std_times, by="row_id")
o2_a1c = o2_a1c %>% mutate(Sex = ifelse(o2_a1c$Sex == "M", 1, 0))
o2_bp_ids = o2_bp$UniqueIdentifier %>% unique() %>% sample(0.5 * 0.05 * nrow(o2_bp))
o2_bp_pdt = o2_bp %>% filter(UniqueIdentifier %in% o2_bp_ids) %>% mutate(Marsh = as.factor(Marsh))
ggplot(data=o2_bp_pdt, mapping=aes(x=std_time, y=Systolic, group=UniqueIdentifier, color=Marsh)) +
  geom_line() +
  geom_point() +
  labs(
    title="Spaghetti Plot of SBP Against Time",
    x="time since pre-measurement (days)",
    y="SBP (mm/Hg)"
  )

o2_a1c_ids = o2_a1c$UniqueIdentifier %>% unique() %>% sample(0.5 * 0.05 * nrow(o2_a1c))
o2_a1c_pdt = o2_a1c %>% filter(UniqueIdentifier %in% o2_a1c_ids) %>% mutate(Marsh = as.factor(Marsh))
ggplot(data=o2_a1c_pdt, mapping=aes(x=std_time, y=A1c, group=UniqueIdentifier, color=Marsh)) +
  geom_line() +
  geom_point() +
  labs(
    title="Spaghetti Plot of A1c Against Time",
    x="time since pre-measurement (days)",
    y="A1c (percentage)"
  )

mod_o2_bp = lme(
  fixed = Systolic ~ std_months * Marsh + age + Sex + IncomeLevel + avg.bmi,
  random = ~ std_months | UniqueIdentifier,
  data = o2_bp,
  method = "REML"
)
summary(mod_o2_bp)

mod_o2_a1c = lme(
  fixed = A1c ~ std_months * Marsh + age + Sex + IncomeLevel + avg.bmi,
  random = ~ std_months | UniqueIdentifier,
  data = o2_a1c,
  method = "REML"
)
summary(mod_o2_a1c)
#####
# format for regression results from models
#####

coef_names = c("std_months:koh_catOne", "std_months:koh_catMultiple", "koh_catOne", "koh_catMultiple",
col_names = c("est.", "lower", "upper")

# primary objective
obj1_res = data.frame(

```

```

rbind(
  cbind(intervals(mod_o1_a1c)$fixed[coef_names, col_names], summary(mod_o1_a1c)$tTable[coef_names, "p-value"],
  cbind(intervals(mod_o1_bp, which="fixed")$fixed[coef_names, col_names], summary(mod_o1_bp)$tTable[coef_names, "p-value"])
)

# obj1_res = data.frame(
#   est = c(summary(mod_o1_a1c)$tTable[coef_names, "Value"], summary(mod_o1_bp)$tTable[coef_names, "Value"]),
#   CI = intervals(mod_o1_a1c)$fixed[coef_names, col_names]
# )

rownames(obj1_res) = c("months:KOHonce", "months:KOHmultiple", "KOHonce", "KOHmultiple", "months", "months")
colnames(obj1_res)[4] = "p-value"

kable(obj1_res) %>% pack_rows("a1c_pre_post", 1, 5) %>% pack_rows("sbp_pre_post", 6, 10)

# secondary objective
coef_names = c("std_months:Marsh", "Marsh", "std_months")
col_names = c("est.", "lower", "upper")

obj1_res = data.frame(
  rbind(
    cbind(intervals(mod_o2_a1c, which="fixed")$fixed[coef_names, col_names], summary(mod_o2_a1c)$tTable[coef_names, "p-value"],
    cbind(intervals(mod_o2_bp, which="fixed")$fixed[coef_names, col_names], summary(mod_o2_bp)$tTable[coef_names, "p-value"])
  )
)

# obj1_res = data.frame(
#   est = c(summary(mod_o1_a1c)$tTable[coef_names, "Value"], summary(mod_o1_bp)$tTable[coef_names, "Value"]),
#   CI = intervals(mod_o1_a1c)$fixed[coef_names, col_names]
# )

rownames(obj1_res) = c("months:Marsh", "Marsh", "months", "months:Marsh ", "Marsh ", "months ")
colnames(obj1_res)[4] = "p-value"

kable(obj1_res) %>% pack_rows("a1c_pre_post", 1, 3) %>% pack_rows("sbp_pre_post", 4, 6)
# table for undadjusted baseline results
o1_a1c_bltb = o1_a1c %>%
  group_by(UniqueIdentifier) %>%
  summarize(
    pre_Date = min(A1cDate),
    A1c = ifelse(A1cDate == pre_Date, A1c, NA),
    koh_cat = koh_cat
  ) %>%
  drop_na()

o1_bp_bltb = o1_bp %>%
  group_by(UniqueIdentifier) %>%
  summarize(
    pre_Date = min(BPDate),
    sys = ifelse(BPDate == pre_Date, sys, NA),
    koh_cat = koh_cat
  ) %>%

```

```

drop_na()

o1_bltb = data.frame(
  KOHnever = c(mean((o1_a1c_bltb %>% filter(koh_cat == "None"))$A1c), mean((o1_bp_bltb %>% filter(koh_cat == "None"))$SBP)),
  KOHonce = c(mean((o1_a1c_bltb %>% filter(koh_cat == "One"))$A1c), mean((o1_bp_bltb %>% filter(koh_cat == "One"))$SBP)),
  KOHmultiple = c(mean((o1_a1c_bltb %>% filter(koh_cat == "Multiple"))$A1c), mean((o1_bp_bltb %>% filter(koh_cat == "Multiple"))$SBP))
)

rownames(o1_bltb) = c("A1c", "SBP")

o1_bltb

```