01_homework1

Randy
1/29/2021

BIOS7721 Homework1

Question1 longitudinal analysis

a. the number of measurements varies

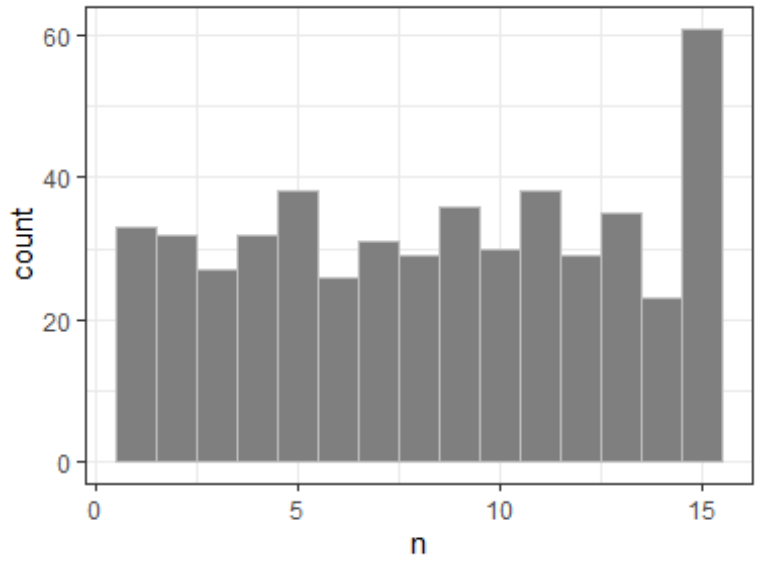
- calculate each subject measurement number
- distribution of aortic gradient
- why consider square root transformation
- create sqrt.aort.gard column for transformation

The dataset is not well balanced for each individual.
The measurement times are calculated and presented in *plot_count*.
The distribution of aortic gradient is presented in *plot_hist_ag*.
According to the *plot_hist_ag*, the distribution of aortic gradient is highly right-skewed. To balance the long right tail, the square root transformation is applied. After the transformation, the sqrt aortic gradient looks more similar to normal distribution.

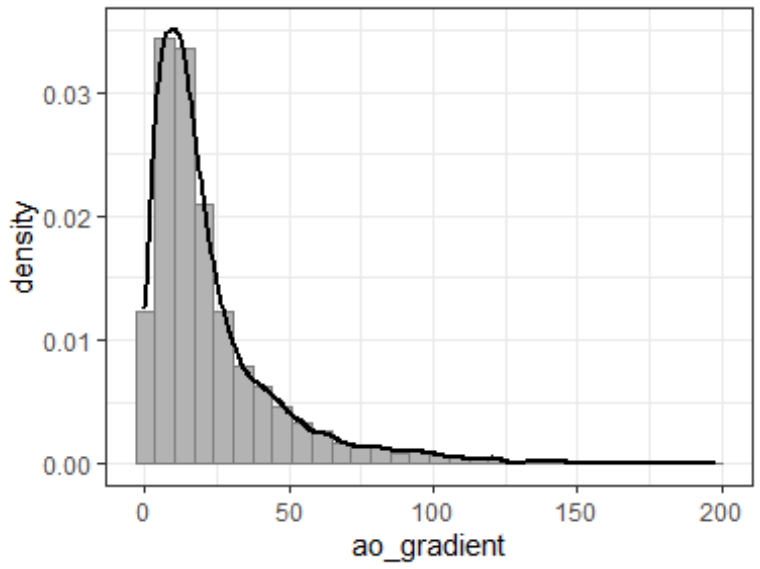
```
aort <- here::here("aort.csv") %>%
  read_csv() %>%
  janitor::clean_names()

## to count each id's measurement
aort_count <- aort %>%
  group_by(id) %>%
  count()
# aort_count

## to get the frequency of measurement
aort_recount <-
  aort_count %>%
  group_by(n) %>%
  count()
# aort_recount
## plot for measurement frequency
plot_count <-
  aort_count %>%
  ggplot(aes(n, fill = "aort")) +
  geom_histogram(binwidth = 1,
                 fill = "grey50",
                 color = "grey") +
  theme(legend.position = "none") +
  theme_bw()
plot_count
```



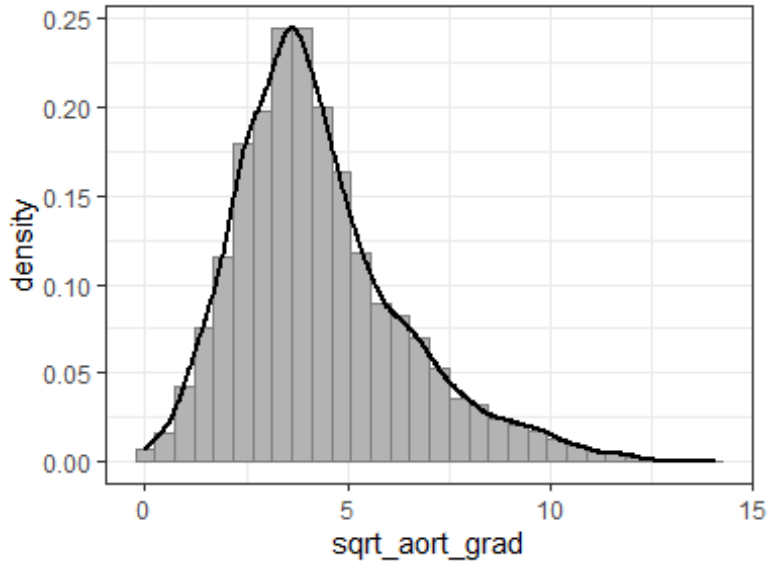

```
## plot for the ag distribution
plot_hist_ag <- aort %>%
  ggplot(aes(ao_gradient)) +
  geom_histogram(aes(y = ..density..),
                 fill = "grey70",
                 color = "grey50") +
  geom_density(alpha = 0.1,
               size = 1) +
  theme(legend.position="none") +
  theme_bw()
plot_hist_ag
```




```
## to add extra sqrt_aort_grad
aort1 <- aort %>%
  mutate(sqrt_aort_grad = sqrt(ao_gradient),
         event = as.factor(event),
         id= as.factor(id))

## after sqrt transformation
```

```
plot_hist_sag <- aort1 %>%
  ggplot(aes(sqrt_aort_grad)) +
  geom_histogram(aes(y = ..density..),
                 fill = "grey70",
                 color = "grey50") +
  geom_density(alpha = 0.1,
               size = 1) +
  theme(legend.position="none") +
  theme_bw()
plot_hist_sag
```




```
head(aort1, 5)
## # A tibble: 5 x 9
##   id    ao_gradient  time ev_time event type_op sex     age sqrt_aort_grad
##   <fct>       <dbl> <dbl>   <dbl> <fct> <chr>   <chr> <dbl>          <dbl>
## 1 1            4.32 0        1.86 0     SI      Male   43.7           2.08
## 2 1            3.65 0.750    1.86 0     SI      Male   43.7           1.91
## 3 2            8.25 0       12.6  0     SI      Male   69.1           2.87
## 4 2            6.43 3.23    12.6  0     SI      Male   69.1           2.54
## 5 2           27.4  3.82    12.6  0     SI      Male   69.1           5.24
```

b. subset 5 subjects

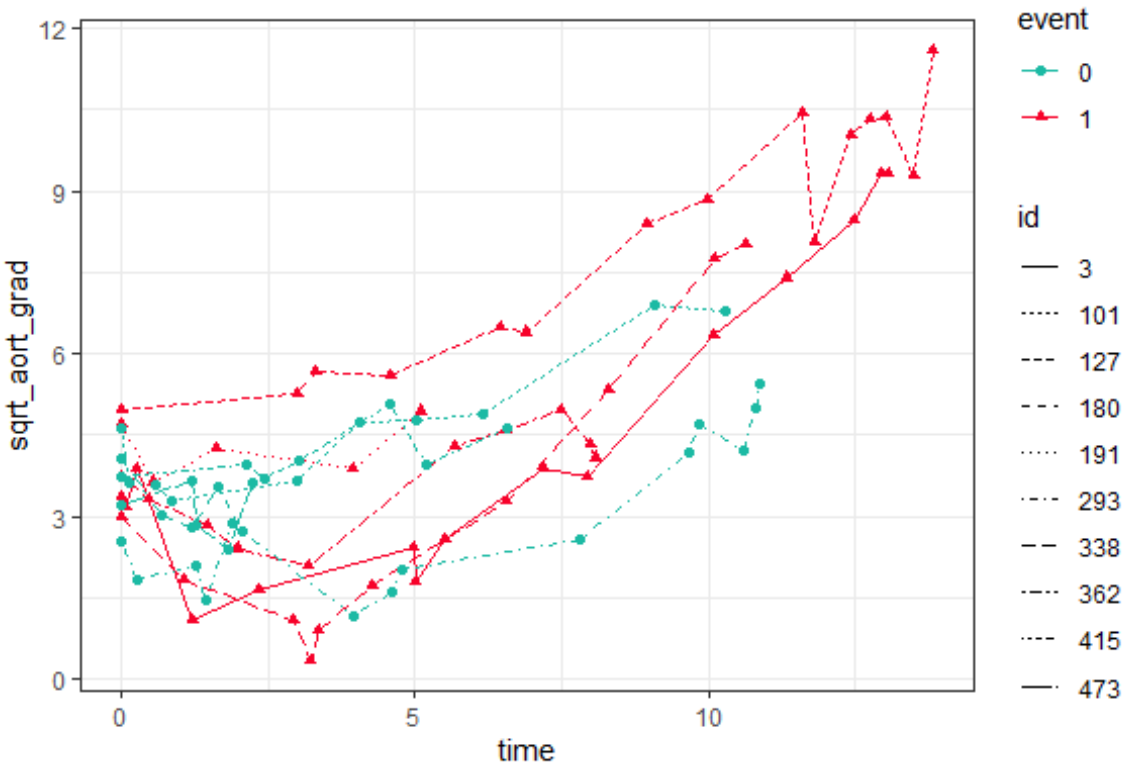
- for with or without events
- plot and describe observed trends

The trend are affected by the samples chosen. Based on these ten samples, the subjects without events have shorter follow-up time; also some of their aortic gradient level increases slower over the time, but the others aortic gradient trends have no difference with the subjects suffered events during follow-up. This indicates potential informative censor or missing not at random cases. Both groups time trend can be approximated through linear or quadratic trend.

```
set.seed(55)
aort_live <- aort1 %>%
  group_by(id) %>%
  filter(event == 0) %>%
  ## to nest the dataset
  ## in one cell
  nest() %>%
  as.tibble() %>%
  sample_n(size = 5) %>%
  unnest()

aort_dead <- aort1 %>%
  group_by(id) %>%
  filter(event == 1) %>%
  ## to nest the dataset
  ## in one cell
  nest() %>%
  as.tibble() %>%
  sample_n(size = 5) %>%
  unnest()
plot_sample <-
  rbind(aort_dead, aort_live) %>%
  ggplot(aes(x = time,
             y = sqrt_aort_grad,
             group = id,
             linetype = id,
             color = event)) +
  geom_line() +
  geom_point(aes(shape = event)) +
  theme_bw() +
  scale_color_manual(values = c("#1cbaa4", "#f7022a"))

plot_sample
```


sqrt_aort_grad
12
9
6
3
0
0
5
10
time
event
0
1
id
3
101
127
180
191
293
338
362
415
473

c. fit a random intercept model

- sqrt_aort_grad as outcome
- linear effect of time
- mean aortic gradient over time?
- change vary be surgery?
- variation of baseline between subject?
- hypothesis test
- interpret the coefficient

According to the random intercept model (model1), the time has a very highly significant effect on the subject's aortic gradient level ($p << 0.001$). Hence, there is an evidence for the mean aortic gradient changing over time.
According to the model adjusted for operation type (model2), there is a very highly significant effect on the subject's aortic gradient level over different operation type ($p << 0.001$); however the time effect on different operation type are not significant ($p = 0.67 > 0.05$).
There is a variability for each subject aortic gradient level.

```
model1 <- lme(sqrt_aort_grad ~ time,
              random = ~1 | id,
              data = aort1)
model2 <- lme(sqrt_aort_grad ~ time * type_op,
              random = ~1 | id,
              data = aort1)

tidy1 <- broom.mixed::tidy(model1)
tidy2 <- broom.mixed::tidy(model2)

glance1 <- broom.mixed::glance(model1)
glance2 <- broom.mixed::glance(model2)
# augment1 <- broom.mixed::augment(model1)
# augment2 <- broom.mixed::augment(model2)
## effects for model1, model2
tidy1 %>% knitr::kable("simple", align = "c")
```

effect	group	term	estimate	std.error	df	statistic	p.value
fixed	fixed	(Intercept)	2.6069753	0.0566041	3682	46.05626	0
fixed	fixed	time	0.3906919	0.0046199	3682	84.56675	0
ran_pars	id	sd_(Intercept)	1.1258056	NA	NA	NA	NA
ran_pars	Residual	sd_Observation	1.0000321	NA	NA	NA	NA

```
tidy2 %>% knitr::kable("simple", align = "c")
```

effect	group	term	estimate	std.error	df	statistic	p.value
fixed	fixed	(Intercept)	2.2305732	0.0786304	3681	28.3678260	0.000000
fixed	fixed	time	0.3925487	0.0064359	3681	60.9936987	0.000000
fixed	fixed	type_opSI	0.7210579	0.1089985	498	6.6153033	0.000000
fixed	fixed	time:type_opSI	-0.0038301	0.0092276	3681	-0.4150671	0.678117
ran_pars	id	sd_(Intercept)	1.0719325	NA	NA	NA	NA
ran_pars	Residual	sd_Observation	0.9999630	NA	NA	NA	NA

To see the subject individual aortic gradient variability to baseline level, the baseline aortic gradient level is adjusted in model3. According to model3, baseline aortic gradient level can significantly affect the subject's future aortic gradient level. These results indicate that there is a variation for baseline aortic gradient. After adjustment on the baseline level, the model performs better.

```
aort2 <- aort1 %>%
  mutate(baseline = ifelse(time == 0, ao_gradient, NA)) %>%
  fill(baseline) %>%
  mutate(basesqrt = sqrt(baseline))

model3 <- lme(sqrt_aort_grad ~ basesqrt + time * type_op,
              random = ~  1 | id,
              data = aort2)

tidyf3 <- broom.mixed::tidy(model3)
tidyr3 <- broom.mixed::tidy(model3, effects = "ran_pars", conf.int = TRUE)
glance3 <- broom.mixed::glance(model3)
tidyf3 %>% knitr::kable("simple", align = "c")
```

effect	group	term	estimate	std.error	df	statistic	p.value
fixed	fixed	(Intercept)	0.0860306	0.1327166	3681	0.648228	0.5168779
fixed	fixed	basesqrt	0.7622855	0.0414871	497	18.374041	0.0000000
fixed	fixed	time	0.3928163	0.0063488	3681	61.872802	0.0000000
fixed	fixed	type_opSI	0.1398787	0.0932231	497	1.500473	0.1341268
fixed	fixed	time:type_opSI	-0.0049038	0.0090914	3681	-0.539387	0.5896525
ran_pars	id	sd_(Intercept)	0.7970539	NA	NA	NA	NA
ran_pars	Residual	sd_Observation	0.9979309	NA	NA	NA	NA

```
tidyr3 %>% knitr::kable("simple", align = "c")
```

effect	group	term	estimate	conf.low	conf.high
ran_pars	id	sd_(Intercept)	0.7970539	0.7399557	0.858558
ran_pars	Residual	sd_Observation	0.9979309	NA	NA

As seen for the criteria comparison cross the three models, model3, adjusting both operation type and the baseline aortic gradients, performs best with lowest AIC (12748) and BIC (12793).

Also based on log-likelihood, the chi-squre and pvalue calculated among nested models. The results indicate that the adjustment for both operation type and the baseline aortic gradients can significantly improve the inference model.

For model3, there is a significant time trend for subject's aortic gradient level ($p << 0.001$). On average, in population level, the subject's sqrt aortic gradient level increase 0.39 (95% CI: 0.38, 0.41) unit in each year. Also the operation type does not significantly affect the patient's aortic gradient level ($p = 0.13$), and there is no significant time:operation interaction effect ($p = 0.59$). Intriguingly, the subject's baseline aortic gradient level can also affect the future ($p << 0.01$). On average, in population level, the patient, with higher baseline aortic gradient, The increasing each unit of baseline sqrt aortic gradient will increase 0.76 (95% CI: 0.68, 0.84) unit in future sqrt aortic gradient level accordingly. As random effects, there is a large standard deviation on sqrt aortic gradient level (sd = 0.78) on individual level. Also the random effects residual standard deviation is pretty large too. These indicate the individual level variation could not be ignored but treated as random effects individually.

```
rbind(glance1,
      glance2,
      glance3) %>%
  rownames_to_column("model") %>%
  knitr::kable("simple", align = "c")
```

model	sigma	logLik	AIC	BIC
1	1.0000321	-6512.524	13033.05	13058.40
2	0.9999630	-6495.179	13002.36	13040.38

3 0.9979309 -6367.431 12748.86 12793.23

```r
#' test_lrt() is to calculate the lrt pvalue
#'
#' @param mod0 is the first model
#' @param mod1 is the second model
#' @param df can be add manually
#' @return pvalue is the lrt pvalue
#' @examples
#' test_lrt(model1, model2)
test_lrt <- function(mod0, mod1, ...){
  A <- logLik(mod0)
  B <- logLik(mod1)
  D <- -2 * (as.numeric(A) - as.numeric(B))
  df = abs(attributes(A)$df - attributes(B)$df)
  pvalue <- pchisq(D, df = df,
                   lower.tail = FALSE)
  return(pvalue)
}

test_lrt(model1, model2, 1)
## [1] 2.930505e-08
test_lrt(model2, model3)
## [1] 1.646453e-57
```

d. extend to random slope model

- how many more parameters
- outcome change over time between subject?

Two more parameters are estimated, the estimate for random slope and its correlation matrix. The random slope on time has standard deviation 0.18, with mean set as zero. In this case, the variability of random time slope effects among subjects cannot be ignored.

After addition of random slope term the AIC decreased 884 to 11864; this indicates the variability on random time slope term for each individual.

```
model4 <- lme(sqrt_aort_grad ~ basesqrt + time * type_op,
              random = ~ time + 1 | id,
              data = aort2)
tidy4 <- broom.mixed::tidy(model4)
glance4 <- broom.mixed::glance(model4)
augment4 <- broom.mixed::augment(model4)
## the estimators for model
tidy4 %>% knitr::kable("simple", align = "c")
```

effect	group	term	estimate	std.error	df	statistic	p.value
fixed	fixed	(Intercept)	0.1338520	0.1042781	3681	1.283606	0.1993606
fixed	fixed	basesqrt	0.7647978	0.0329209	497	23.231377	0.0000000
fixed	fixed	time	0.3542383	0.0151700	3681	23.351252	0.0000000
fixed	fixed	type_opSI	0.1994582	0.0727786	497	2.740616	0.0063532
fixed	fixed	time:type_opSI	-0.0376092	0.0213554	3681	-1.761115	0.0783019
ran_pars	id	sd_(Intercept)	0.5658046	NA	NA	NA	NA
ran_pars	id	cor_time.(Intercept)	-0.1588935	NA	NA	NA	NA
ran_pars	id	sd_time	0.1794941	NA	NA	NA	NA
ran_pars	Residual	sd_Observation	0.8417204	NA	NA	NA	NA

```
rbind(glance1,
      glance2,
      glance3,
      glance4) %>%
  tibble() %>%
  rownames_to_column("model") %>%
  knitr::kable("simple", align = "c")
```

model	sigma	logLik	AIC	BIC
1	1.0000321	-6512.524	13033.05	13058.40
2	0.9999630	-6495.179	13002.36	13040.38
3	0.9979309	-6367.431	12748.86	12793.23
4	0.8417204	-5923.191	11864.38	11921.42

e. explore splines in fixed and random effects

- comments on results

Because the time:operation interaction term did not contribute to the model performance, the interaction term is removed from model6. After remove the interaction, there is no significant improvement for model6 (AIC = 9893) than model5 (AIC = 9899).

The B-spline adds more flexibility to the model, so the over all fitting performs much better (with much lower AIC). The time trend and operation type can all significantly affect the aortic gradient level; but the time:operation interaction terms are still not significant, which is consistent for model5 and model6. The overall results are pretty consistent with earlier models.

```
model5 <- lme(sqrt_aort_grad ~ bs(time) * type_op,
              random = list(id = pdDiag(form = ~ bs(time))),
              ## if the entire matrix used
              ## then there might be a numerical problem
              data = aort2)

model6 <- lme(sqrt_aort_grad ~ bs(time) + type_op,
              random = list(id = pdDiag(form = ~ bs(time))),
              ## if the entire matrix used
              ## then there might be a numerical problem
              data = aort2)

tidy5 <- broom.mixed::tidy(model5, effects = "fixed")
glance5 <- broom.mixed::glance(model5)
augment5 <- broom.mixed::augment(model5)
tidy6 <- broom.mixed::tidy(model6, effects = "fixed")
glance6 <- broom.mixed::glance(model6)
augment6 <- broom.mixed::augment(model6)

summary(model5)
## Linear mixed-effects model fit by REML
##  Data: aort2
##        AIC      BIC    logLik
##   9899.758 9982.138 -4936.879
##
## Random effects:
##  Formula: ~bs(time) | id
##  Structure: Diagonal
##         (Intercept) bs(time)1 bs(time)2 bs(time)3  Residual
## StdDev:   0.7453031  2.170479  2.633737  2.269301 0.5753696
##
## Fixed effects: sqrt_aort_grad ~ bs(time) * type_op
##                           Value Std.Error   DF  t-value p-value
## (Intercept)           2.887873 0.0570636 3677 50.60800  0.0000
## bs(time)1            -0.594246 0.2032711 3677 -2.92342  0.0035
## bs(time)2             2.648580 0.2921029 3677  9.06728  0.0000
## bs(time)3             8.239182 0.3293511 3677 25.01641  0.0000
## type_opSI             0.688341 0.0788852  498  8.72585  0.0000
## bs(time)1:type_opSI   0.002985 0.2848500 3677  0.01048  0.9916
## bs(time)2:type_opSI  -0.024445 0.4154472 3677 -0.05884  0.9531
## bs(time)3:type_opSI  -0.239899 0.4753256 3677 -0.50470  0.6138
##  Correlation:
##                      (Intr) bs(t)1 bs(t)2 bs(t)3 typ_SI b()1:_ b()2:_
## bs(time)1            -0.254
## bs(time)2             0.074 -0.348
## bs(time)3            -0.115  0.278 -0.450
## type_opSI            -0.723  0.184 -0.054  0.083
## bs(time)1:type_opSI   0.182 -0.714  0.248 -0.199 -0.252
## bs(time)2:type_opSI  -0.052  0.245 -0.703  0.317  0.075 -0.361
## bs(time)3:type_opSI   0.080 -0.193  0.312 -0.693 -0.110  0.274 -0.450
```

```
##
## Standardized Within-Group Residuals:
##         Min          Q1         Med          Q3         Max
## -3.14282961 -0.58596222  0.01429144  0.58172863  3.36845306
##
## Number of Observations: 4183
## Number of Groups: 500
summary(model6)
## Linear mixed-effects model fit by REML
##  Data: aort2
##        AIC      BIC    logLik
##   9893.488 9956.863 -4936.744
##
## Random effects:
##  Formula: ~bs(time) | id
##  Structure: Diagonal
##         (Intercept) bs(time)1 bs(time)2 bs(time)3 Residual
## StdDev:   0.7452915  2.167305  2.630314  2.256675 0.575376
##
## Fixed effects: sqrt_aort_grad ~ bs(time) + type_op
##                 Value  Std.Error   DF  t-value p-value
## (Intercept)  2.889270 0.05606046 3680 51.53847       0
## bs(time)1   -0.594253 0.14223484 3680 -4.17797       0
## bs(time)2    2.638427 0.20743757 3680 12.71914       0
## bs(time)3    8.120202 0.23658423 3680 34.32267       0
## type_opSI    0.685947 0.07620478  498  9.00137       0
##  Correlation:
##           (Intr) bs(t)1 bs(t)2 bs(t)3
## bs(time)1 -0.181
## bs(time)2  0.055 -0.362
## bs(time)3 -0.083  0.275 -0.451
## type_opSI -0.711  0.005 -0.004  0.008
##
## Standardized Within-Group Residuals:
##         Min          Q1         Med          Q3         Max
## -3.14269745 -0.58752588  0.01504345  0.58305989  3.36704422
##
## Number of Observations: 4183
## Number of Groups: 500
tidy5 %>% knitr::kable("simple", align = "c")
```

term	estimate	std.error	df	statistic	p.value
(Intercept)	2.8878727	0.0570636	3677	50.6079964	0.0000000
bs(time)1	-0.5942460	0.2032711	3677	-2.9234162	0.0034833
bs(time)2	2.6485799	0.2921029	3677	9.0672835	0.0000000
bs(time)3	8.2391819	0.3293511	3677	25.0164131	0.0000000
type_opSI	0.6883405	0.0788852	498	8.7258473	0.0000000
bs(time)1:type_opSI	0.0029854	0.2848500	3677	0.0104807	0.9916384
bs(time)2:type_opSI	-0.0244449	0.4154472	3677	-0.0588400	0.9530828
bs(time)3:type_opSI	-0.2398991	0.4753256	3677	-0.5047048	0.6137964

```
tidy6 %>% knitr::kable("simple", align = "c")
```

term	estimate	std.error	df	statistic	p.value
(Intercept)	2.8892703	0.0560605	3680	51.538473	0.00e+00
bs(time)1	-0.5942528	0.1422348	3680	-4.177969	3.01e-05
bs(time)2	2.6384272	0.2074376	3680	12.719139	0.00e+00
bs(time)3	8.1202024	0.2365842	3680	34.322669	0.00e+00
type_opSI	0.6859475	0.0762048	498	9.001371	0.00e+00

```
rbind(glance1,
      glance2,
      glance3,
      glance4,
      glance5,
      glance6) %>%
```

```
  tibble() %>%
  rownames_to_column("model") %>%
  knitr::kable("simple", align = "c")
```

model	sigma	logLik	AIC	BIC
1	1.0000321	-6512.524	13033.049	13058.402
2	0.9999630	-6495.179	13002.358	13040.385
3	0.9979309	-6367.431	12748.863	12793.226
4	0.8417204	-5923.191	11864.381	11921.419
5	0.5753696	-4936.879	9899.758	9982.138
6	0.5753760	-4936.744	9893.488	9956.863

f. plot the population trajectory

- plot the predict 10 patients trajectory

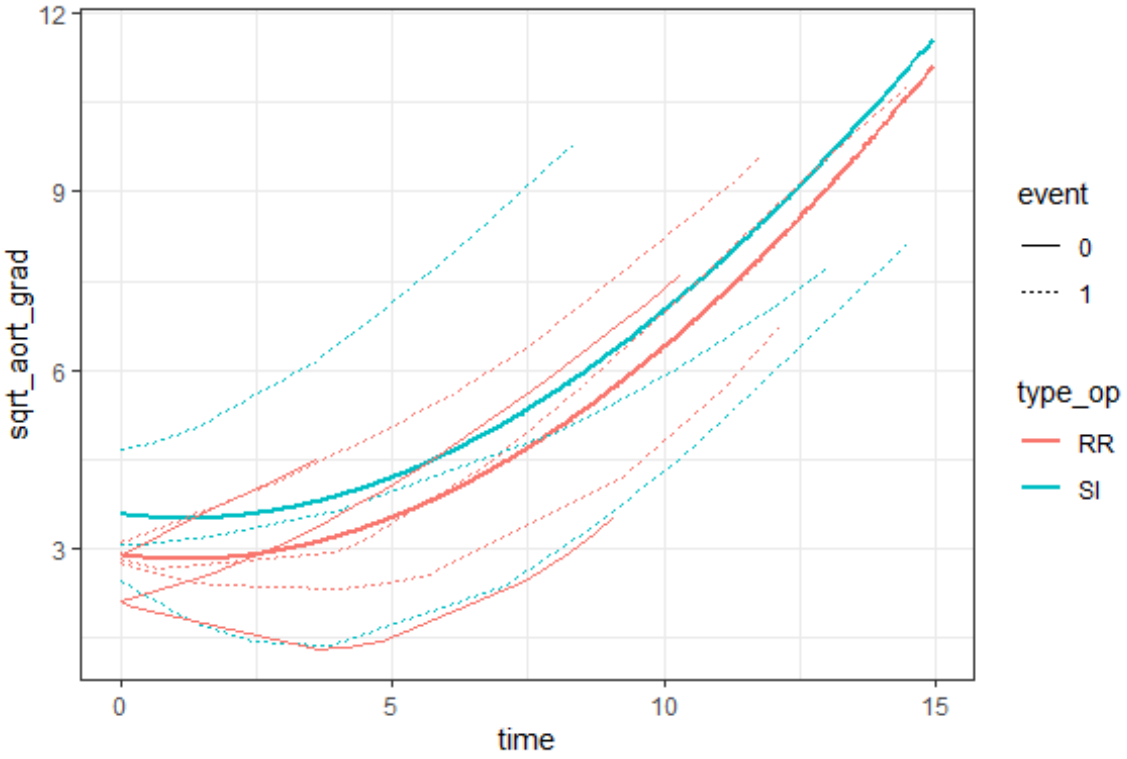
Over all the trajectory can be approximated through quadratic or cubic function as increasing pattern.
In population level, the SI operation has a higher aortic gradient level than the RR operation. There is a really high variability on the random effect on subject level. Also the subjects did not suffer events have shorter follow-up time, even the trends seem similar to the subjects with event.

There is a chance that the missing mechanism is not random; without further analysis on the missing mechanism or data implement, the results might be biased.

```
set.seed(5555)
augment5 <- broom.mixed::augment(model5) %>%
  as.data.frame()

## use level = 0 for the population
augment5_sub <-
  augment5 %>%
  group_by(id) %>%
  nest() %>%
  tibble() %>%
  sample_n(size = 10,
           replace = FALSE) %>%
  unnest() %>%
  tibble()
plot_model5 <-
  ggplot(augment5) +
  geom_line(aes(x = time,
                y = .fixed,
                group = type_op,
                color = type_op),
            linetype = "solid",
            size = 1) +
  geom_line(data = augment5_sub,
            aes(x = time,
                y = .fitted,
                group = id,
                linetype = event,
                color = type_op)) +
  xlab("time") +
  ylab("sqrt_aort_grad") +
  theme_bw()

plot_model5
```


12
9
6
3
sqrt_aort_grad
0
5
10
15
time
event
0
1
type_op
RR
SI

g. variance-covariance matrix for patient

- correlation matrix for patient
- from model2, model3, and model4

For model2 the covariance and correlation matrices are positive definite, the correlation keeps the same cross the time.
For model3 the covariance and correlation matrices are still positive definite and symmetric, but the correlation decreases over time.
For model5 the covariance and correlation matrices are still positive definite and symmetric, but there is no obvious pattern over time.

```
#' get_cov_cor() is for covariance and correlation
#'
#' @param object a lm or lmm object
#' @param id the subjec interested with
#' @param type options: "margin", "residual", and "condition"
#' @return mcov(marginal covariance matrix)
#' @return cor(correlation matrix based on mcov)
#' @examples
#' get_cov_cor(model1, 5, "condition")
get_cov_cor <-
  function(object, id, type = "margin") {
    mcov <- nlme::getVarCov(
            obj = object,
            individual = id,
            type = type)
    cor <- stats::cov2cor(mcov[[1]]) %>%
      round(2)

    mcov <- mcov[["2"]] %>%
      round(2)
    return(list(mcov, cor))
  }
get_cov_cor(model2, 2) %>% knitr::kable("simple", align = "c")
```

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15
1.15	2.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15
1.15	1.15	2.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15
1.15	1.15	1.15	2.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15
1.15	1.15	1.15	1.15	2.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15
1.15	1.15	1.15	1.15	1.15	2.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15
1.15	1.15	1.15	1.15	1.15	1.15	2.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15
1.15	1.15	1.15	1.15	1.15	1.15	1.15	2.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15
1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	2.15	1.15	1.15	1.15	1.15	1.15	1.15
1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	2.15	1.15	1.15	1.15	1.15	1.15
1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	2.15	1.15	1.15	1.15	1.15
1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	2.15	1.15	1.15	1.15
1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	2.15	1.15	1.15
1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	2.15	1.15
1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	2.15

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1.00	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53
0.53	1.00	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53
0.53	0.53	1.00	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53
0.53	0.53	0.53	1.00	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53

0.53	0.53	0.53	0.53	1.00	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53
0.53	0.53	0.53	0.53	0.53	1.00	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53
0.53	0.53	0.53	0.53	0.53	0.53	1.00	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53
0.53	0.53	0.53	0.53	0.53	0.53	0.53	1.00	0.53	0.53	0.53	0.53	0.53	0.53	0.53
0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	1.00	0.53	0.53	0.53	0.53	0.53	0.53
0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	1.00	0.53	0.53	0.53	0.53	0.53
0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	1.00	0.53	0.53	0.53	0.53
0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	1.00	0.53	0.53	0.53
0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	1.00	0.53	0.53
0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	1.00	0.53
0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	1.00

```
get_cov_cor(model4, 2) %>% knitr::kable("simple", align = "c")
```

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1.03	0.27	0.26	0.24	0.23	0.23	0.21	0.21	0.19	0.18	0.17	0.16	0.15	0.14	0.13
0.27	1.26	0.60	0.68	0.74	0.76	0.85	0.87	0.96	1.04	1.10	1.13	1.22	1.24	1.32
0.26	0.60	1.38	0.76	0.83	0.86	0.97	1.00	1.10	1.20	1.27	1.31	1.41	1.44	1.54
0.24	0.68	0.76	1.59	0.97	1.01	1.14	1.18	1.31	1.43	1.52	1.57	1.70	1.74	1.87
0.23	0.74	0.83	0.97	1.78	1.11	1.27	1.31	1.47	1.60	1.71	1.77	1.92	1.96	2.11
0.23	0.76	0.86	1.01	1.11	1.87	1.32	1.37	1.53	1.67	1.78	1.85	2.00	2.05	2.21
0.21	0.85	0.97	1.14	1.27	1.32	2.23	1.58	1.77	1.95	2.08	2.15	2.34	2.40	2.59
0.21	0.87	1.00	1.18	1.31	1.37	1.58	2.34	1.84	2.01	2.15	2.23	2.43	2.49	2.68
0.19	0.96	1.10	1.31	1.47	1.53	1.77	1.84	2.78	2.28	2.44	2.53	2.76	2.83	3.05
0.18	1.04	1.20	1.43	1.60	1.67	1.95	2.01	2.28	3.22	2.69	2.79	3.04	3.12	3.37
0.17	1.10	1.27	1.52	1.71	1.78	2.08	2.15	2.44	2.69	3.59	2.99	3.26	3.35	3.62
0.16	1.13	1.31	1.57	1.77	1.85	2.15	2.23	2.53	2.79	2.99	3.82	3.39	3.48	3.76
0.15	1.22	1.41	1.70	1.92	2.00	2.34	2.43	2.76	3.04	3.26	3.39	4.41	3.80	4.11
0.14	1.24	1.44	1.74	1.96	2.05	2.40	2.49	2.83	3.12	3.35	3.48	3.80	4.61	4.22
0.13	1.32	1.54	1.87	2.11	2.21	2.59	2.68	3.05	3.37	3.62	3.76	4.11	4.22	5.28

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1.00	0.24	0.22	0.19	0.17	0.17	0.14	0.13	0.11	0.10	0.09	0.08	0.07	0.07	0.05
0.24	1.00	0.46	0.48	0.49	0.50	0.51	0.51	0.51	0.52	0.52	0.52	0.52	0.51	0.51
0.22	0.46	1.00	0.52	0.53	0.54	0.55	0.56	0.56	0.57	0.57	0.57	0.57	0.57	0.57
0.19	0.48	0.52	1.00	0.58	0.58	0.61	0.61	0.62	0.63	0.64	0.64	0.64	0.64	0.65
0.17	0.49	0.53	0.58	1.00	0.61	0.64	0.64	0.66	0.67	0.68	0.68	0.68	0.69	0.69
0.17	0.50	0.54	0.58	0.61	1.00	0.65	0.65	0.67	0.68	0.69	0.69	0.70	0.70	0.70
0.14	0.51	0.55	0.61	0.64	0.65	1.00	0.69	0.71	0.73	0.73	0.74	0.75	0.75	0.75
0.13	0.51	0.56	0.61	0.64	0.65	0.69	1.00	0.72	0.73	0.74	0.75	0.76	0.76	0.76
0.11	0.51	0.56	0.62	0.66	0.67	0.71	0.72	1.00	0.76	0.77	0.78	0.79	0.79	0.80
0.10	0.52	0.57	0.63	0.67	0.68	0.73	0.73	0.76	1.00	0.79	0.80	0.81	0.81	0.82
0.09	0.52	0.57	0.64	0.68	0.69	0.73	0.74	0.77	0.79	1.00	0.81	0.82	0.82	0.83
0.08	0.52	0.57	0.64	0.68	0.69	0.74	0.75	0.78	0.80	0.81	1.00	0.83	0.83	0.84
0.07	0.52	0.57	0.64	0.68	0.70	0.75	0.76	0.79	0.81	0.82	0.83	1.00	0.84	0.85
0.07	0.51	0.57	0.64	0.69	0.70	0.75	0.76	0.79	0.81	0.82	0.83	0.84	1.00	0.86
0.05	0.51	0.57	0.65	0.69	0.70	0.75	0.76	0.80	0.82	0.83	0.84	0.85	0.86	1.00

```
get_cov_cor(model5, 2) %>% knitr::kable("simple", align = "c")
```

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0.89	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56
0.56	1.72	1.46	1.54	1.58	1.58	1.58	1.58	1.53	1.46	1.39	1.35	1.23	1.19	1.05
0.56	1.46	1.89	1.65	1.69	1.71	1.72	1.72	1.67	1.61	1.54	1.49	1.37	1.32	1.17
0.56	1.54	1.65	2.10	1.84	1.85	1.89	1.90	1.87	1.82	1.76	1.71	1.59	1.54	1.37

0.56	1.58	1.69	1.84	2.24	1.94	2.00	2.00	2.00	1.96	1.91	1.87	1.74	1.70	1.53
0.56	1.58	1.71	1.85	1.94	2.29	2.03	2.04	2.05	2.02	1.97	1.93	1.81	1.76	1.59
0.56	1.58	1.72	1.89	2.00	2.03	2.47	2.16	2.21	2.20	2.17	2.15	2.05	2.00	1.84
0.56	1.58	1.72	1.90	2.00	2.04	2.16	2.52	2.24	2.24	2.22	2.20	2.10	2.06	1.90
0.56	1.53	1.67	1.87	2.00	2.05	2.21	2.24	2.66	2.38	2.38	2.37	2.31	2.28	2.15
0.56	1.46	1.61	1.82	1.96	2.02	2.20	2.24	2.38	2.78	2.48	2.49	2.47	2.45	2.35
0.56	1.39	1.54	1.76	1.91	1.97	2.17	2.22	2.38	2.48	2.87	2.56	2.57	2.57	2.50
0.56	1.35	1.49	1.71	1.87	1.93	2.15	2.20	2.37	2.49	2.56	2.92	2.62	2.62	2.59
0.56	1.23	1.37	1.59	1.74	1.81	2.05	2.10	2.31	2.47	2.57	2.62	3.05	2.74	2.77
0.56	1.19	1.32	1.54	1.70	1.76	2.00	2.06	2.28	2.45	2.57	2.62	2.74	3.10	2.83
0.56	1.05	1.17	1.37	1.53	1.59	1.84	1.90	2.15	2.35	2.50	2.59	2.77	2.83	3.31

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1.00	0.45	0.43	0.41	0.39	0.39	0.38	0.37	0.36	0.35	0.35	0.35	0.34	0.34	0.32
0.45	1.00	0.81	0.81	0.80	0.80	0.77	0.76	0.71	0.67	0.63	0.60	0.54	0.51	0.44
0.43	0.81	1.00	0.83	0.82	0.82	0.80	0.79	0.75	0.70	0.66	0.64	0.57	0.55	0.47
0.41	0.81	0.83	1.00	0.84	0.84	0.83	0.82	0.79	0.75	0.72	0.69	0.63	0.60	0.52
0.39	0.80	0.82	0.84	1.00	0.85	0.85	0.84	0.82	0.79	0.75	0.73	0.67	0.64	0.56
0.39	0.80	0.82	0.84	0.85	1.00	0.85	0.85	0.83	0.80	0.77	0.75	0.68	0.66	0.58
0.38	0.77	0.80	0.83	0.85	0.85	1.00	0.87	0.86	0.84	0.82	0.80	0.74	0.72	0.64
0.37	0.76	0.79	0.82	0.84	0.85	0.87	1.00	0.86	0.85	0.83	0.81	0.76	0.74	0.66
0.36	0.71	0.75	0.79	0.82	0.83	0.86	0.86	1.00	0.87	0.86	0.85	0.81	0.79	0.72
0.35	0.67	0.70	0.75	0.79	0.80	0.84	0.85	0.87	1.00	0.88	0.87	0.85	0.83	0.78
0.35	0.63	0.66	0.72	0.75	0.77	0.82	0.83	0.86	0.88	1.00	0.88	0.87	0.86	0.81
0.35	0.60	0.64	0.69	0.73	0.75	0.80	0.81	0.85	0.87	0.88	1.00	0.88	0.87	0.83
0.34	0.54	0.57	0.63	0.67	0.68	0.74	0.76	0.81	0.85	0.87	0.88	1.00	0.89	0.87
0.34	0.51	0.55	0.60	0.64	0.66	0.72	0.74	0.79	0.83	0.86	0.87	0.89	1.00	0.88
0.32	0.44	0.47	0.52	0.56	0.58	0.64	0.66	0.72	0.78	0.81	0.83	0.87	0.88	1.00

Question2 survival analysis

- use the same dataset
- relationships between baseline and survival
- create an individual level dataset

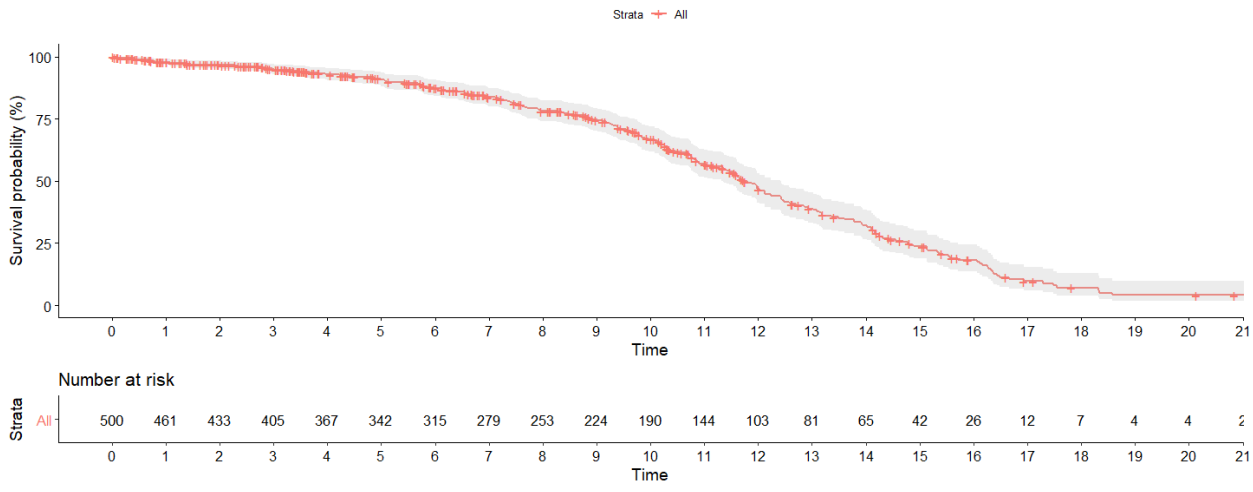
a. Kaplan-Meier survival curves

- overall and operation type

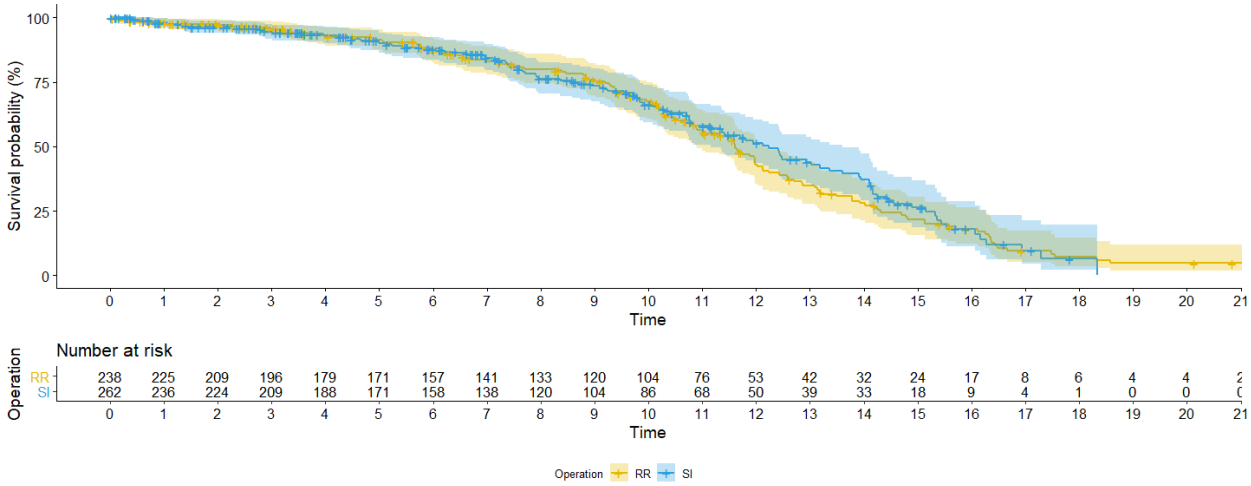
```
aort1 <- filter(aort1, time == 0)
aort1$surobj <- with(aort1, Surv(ev_time, event == 1))

fit0 <- survfit(surobj ~ 1,
                data = aort1,
                type = "kaplan-meier")
fit1 <- survfit(surobj ~ type_op,
                data = aort1,
                type = "kaplan-meier")
plot_fit0 <- fit0 %>%
  survminer::ggsurvplot(
    data = aort1,
    break.time.by = 1,
    conf.int = TRUE,
    fun = "pct",
    risk.table = TRUE,
    size = 1)

plot_fit1 <- fit1 %>%
  survminer::ggsurvplot(
    data = aort1,
    break.time.by = 1,
    conf.int = TRUE,
    fun = "pct",
    risk.table = TRUE,
    size = 1,
    palette= c("#E7B800",
               "#2E9FDF"),
    legend = "bottom",
    legend.title = "Operation",
    legend.labs = c("RR", "SI"))
plot_fit0
```




```
plot_fit1
```


Survival probability (%)
100
75
50
25
0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
Time
Number at risk
Operation
RR 238 225 209 196 179 171 157 141 133 120 104 76 53 42 32 24 17 8 6 4 4
SI 262 236 224 209 188 171 158 138 120 104 86 68 50 39 33 18 9 4 1 0 0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
Time
Operation RR SI

b. result from last question

- estimate the predicted survival at 10 years for each type
- approximate the estimate of the hazard ratio

For operation RR, the estimate survival at 10 years is 0.674; for operation IS, the estimate survival at 10 years is 0.662. So based on the approximation for median survival time and the survival at 10 years.
The proportional hazard ratio for RR and SI is around 1.2 at 10 years.

```
summary(fit1, times = 10)
## Call: survfit(formula = surobj ~ type_op, data = aort1, type = "kaplan-meier")
##
##                 type_op=RR
##         time       n.risk      n.event     survival      std.err lower 95% CI
##      10.0000     104.0000      58.0000       0.6742       0.0358       0.6076
## upper 95% CI
##       0.7482
##
##                 type_op=SI
##         time       n.risk      n.event     survival      std.err lower 95% CI
##      10.0000      86.0000      59.0000       0.6619       0.0373       0.5927
## upper 95% CI
##       0.7391
summary(fit1, times = 9.5)
## Call: survfit(formula = surobj ~ type_op, data = aort1, type = "kaplan-meier")
##
##                 type_op=RR
##         time       n.risk      n.event     survival      std.err lower 95% CI
##       9.5000     110.0000      53.0000       0.7065       0.0345       0.6421
## upper 95% CI
##       0.7775
##
##                 type_op=SI
##         time       n.risk      n.event     survival      std.err lower 95% CI
##       9.5000      99.0000      52.0000       0.7144       0.0345       0.6499
## upper 95% CI
##       0.7854
summary(fit1, times = 10.5)
## Call: survfit(formula = surobj ~ type_op, data = aort1, type = "kaplan-meier")
##
##                 type_op=RR
##         time       n.risk      n.event     survival      std.err lower 95% CI
##      10.5000      87.0000      68.0000       0.6066       0.0381       0.5364
## upper 95% CI
##       0.6861
##
##                 type_op=SI
##         time       n.risk      n.event     survival      std.err lower 95% CI
##      10.5000      78.0000      63.0000       0.6304       0.0387       0.5590
## upper 95% CI
##       0.7110
# survdiff(surobj ~ type_op, data = aort1)
h_rr <- (log(0.7065) - log(0.6066)) / (9.5 - 10.5)
h_si <- (log(0.7144) - log(0.6304)) / (9.5 - 10.5)
h_rr / h_si
## [1] 1.218766
```

c. categorize the baseline value into 4 group

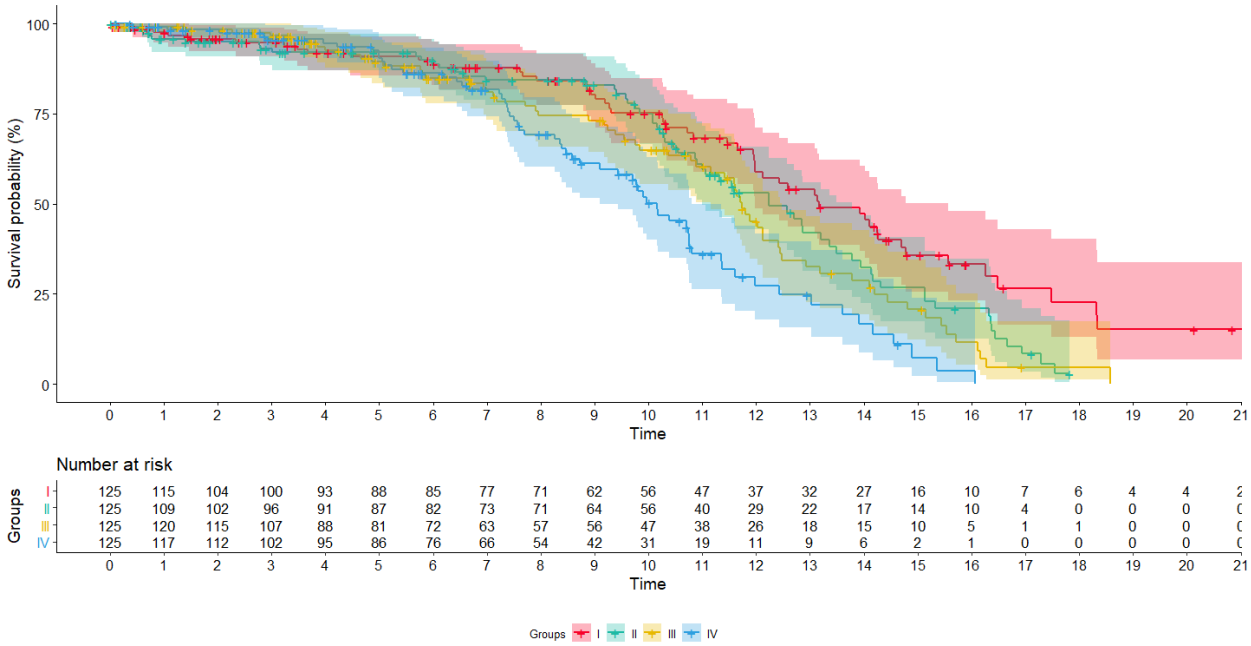
According to the Kaplan-Meier plot (plot_fit2), there is a clear correlation between the baseline aortic gradient level and the survival results. In population level, the lower baseline gradient the subject suffered, the lower risk rate and the longer of the median survival time.

```
#' show_table() to get a frequency table
#'
#' @param data the dataset
#' @param arg the categorical variable interested
#' @return NULL but print function
#' @examples
#' show_table(aort1, sex)
show_table <- function(data, arg) {
  table <- with(data, table(arg)) %>%
    as.data.frame()
  print(table)
}

aort3 <- aort %>%
  mutate(baseline = ifelse(time == 0, ao_gradient, NA)) %>%
  fill(baseline) %>%
  group_by(id, baseline) %>%
  nest() %>%
  mutate(group = case_when(
    baseline <= quantile(.$baseline, 0.25) ~ "1",
    baseline <= quantile(.$baseline, 0.50) ~ "2",
    baseline <= quantile(.$baseline, 0.75) ~ "3",
    baseline <= quantile(.$baseline, 1.00) ~ "4")) %T>%
  show_table(.$group) %>%
  unnest()
##   arg Freq
## 1   1  125
## 2   2  125
## 3   3  125
## 4   4  125
aort3 <- filter(aort3, time == 0)
aort3$surobj <- with(aort3, Surv(ev_time, event == 1))

fit2 <- survfit(surobj ~ group,
                data = aort3)

plot_fit2 <- fit2 %>%
  survminer::ggsurvplot(
    data = aort3,
    break.time.by = 1,
    conf.int = TRUE,
    fun = "pct",
    risk.table = TRUE,
    size = 1,
    palette= c("#f7022a",
               "#1cbaa4",
               "#E7B800",
               "#2E9FDF"),
    legend = "bottom",
    legend.title = "Groups",
    legend.labs = c("I", "II", "III", "IV"))
plot_fit2
```




```
fit2
## Call: survfit(formula = surobj ~ group, data = aort3)
##
##           n events median 0.95LCL 0.95UCL
## group=1 125     55   13.2    12.0    15.6
## group=2 125     63   12.2    11.0    13.8
## group=3 125     63   11.7    11.2    12.5
## group=4 125     62   10.2     9.1    10.8
```

d. fit a cox model and only use operation

Based on the Cox proportional hazard model (cox1), the operation type does not have a significant effect on the patient's survival time (p = 0.68). We cannot reject the null hypothesis. According to the model, the log risk rate will decrease 0.037, which is equivalent to 0.9482 (95% CI: 0.7341, 1.225) fold lower risk, for patient in SI operation than the RR operation.

```
aort1 <- aort1 %>% filter(time == 0)
cox1 <- coxph(surobj ~ type_op,
              data = aort1)
summary(cox1)
## Call:
## coxph(formula = surobj ~ type_op, data = aort1)
##
##   n= 500, number of events= 243
##
##               coef exp(coef) se(coef)      z Pr(>|z|)
## type_opSI -0.05319   0.94820  0.13055 -0.407    0.684
##
##           exp(coef) exp(-coef) lower .95 upper .95
## type_opSI    0.9482      1.055    0.7341     1.225
##
## Concordance= 0.505  (se = 0.019 )
## Likelihood ratio test= 0.17  on 1 df,   p=0.7
## Wald test            = 0.17  on 1 df,   p=0.7
## Score (logrank) test = 0.17  on 1 df,   p=0.7
tidy(cox1) %>% knitr::kable("simple", align = "c")
```

term	estimate	std.error	statistic	p.value
type_opSI	-0.0531859	0.1305531	-0.4073888	0.6837225

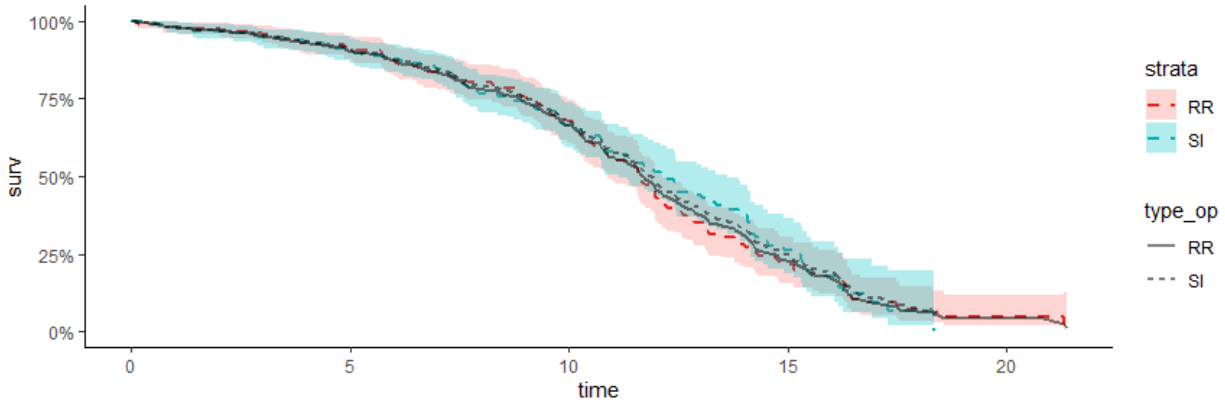
e. predict the models cox1 and fit1

As seen in the overlay plot (plot_cox), the colored dashed lines represent the Kaplan-Meier curves, and the grey solid lines represent the Cox predict curves. Due to no cross-over for different operation types and seemingly constant decreasing on survival rate. The Cox model is reasonable to be used for the model fitting. However due to the constant hazard ratio assumption for Cox model, at particular period of time, certain violations on the assumption for different operational type might happened.

```
aort4 <- aort1 %>%
  mutate(coxsur = predict(cox1, type = "survival"))

plot_cox <-
  ## KM curve plotted by autoplot
  autoplot(fit1,
           data = aort1,
           surv.size = 1,
           surv.linetype = "dashed",
           censor = FALSE) +
  ## add predict cox model values
  geom_line(data = aort4,
            aes(ev_time,
                coxsur,
                group = type_op,
                linetype = type_op),
            color = "black",
            alpha = 0.50,
            size = 1) +
  scale_colour_hue(l = 45, c = 200) +
  theme_classic()

plot_cox
```



f. add baseline value of sqrt_aort_grad

- add base_sqrt_aort into cox model

According to model cox2, both operation types and the baseline aortic gradient levels can significantly affect the subject survival time (pvalue << 0.001). In population level, on average, patients with the SI operation will experience 0.437 decreasing on log risk rate than RR operation; in another words, the risk rate will decrease to 0.646 (95% CI: 0.485, 0.859) fold in SI patients than RR patients.

SI patients can enjoy a better survival time. Also, on average, every unit increase on baseline sqrt aortic gradient level will cause increase of risk rate to 1.567 (95% CI: 1.363, 1.802) folds. This indicates that the patients with higher baseline aortic gradient suffer higher risk of event.

```
aort5 <- aort3 %>%
  mutate(basesqrt = sqrt(baseline))

cox2 <- coxph(surobj ~ type_op + basesqrt,
              data = aort5)
tidyc2 <- broom.mixed::tidy(cox2)
summary(cox2)
## Call:
## coxph(formula = surobj ~ type_op + basesqrt, data = aort5)
##
##   n= 500, number of events= 243
##
##               coef exp(coef) se(coef)      z Pr(>|z|)
## type_opSI -0.43725   0.64581  0.14571 -3.001  0.00269 **
## basesqrt   0.44930   1.56721  0.07122  6.309 2.81e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##           exp(coef) exp(-coef) lower .95 upper .95
## type_opSI    0.6458     1.5484    0.4854    0.8593
## basesqrt     1.5672     0.6381    1.3630    1.8020
##
## Concordance= 0.591  (se = 0.023 )
## Likelihood ratio test= 40.74  on 2 df,   p=1e-09
## Wald test            = 39.86  on 2 df,   p=2e-09
## Score (logrank) test = 39.79  on 2 df,   p=2e-09
tidyc2 %>% knitr::kable("simple", align = "c")
```

term	estimate	std.error	statistic	p.value
type_opSI	-0.4372536	0.1457137	-3.000772	0.002693
basesqrt	0.4492995	0.0712185	6.308744	0.000000

```
glance_c1 <- glance(cox1)
glance_c2 <- glance(cox2)

rbind(glance_c1, glance_c2) %>%
  select(AIC, BIC, logLik) %>%
  rownames_to_column("models") %>%
  knitr::kable("simple", align = "c")
```

models	AIC	BIC	logLik
1	2397.787	2401.280	-1197.894
2	2359.210	2366.196	-1177.605