

1000m Time-Trial and Yo-YoIR1: Reanalysis of Clancy et al.

Mladen Jovanović

11/14/2020

Introduction

This is a re-analysis of the data from the (2) paper. The aim is to provide different analysis, but also educational. I will use functions of the **bmbstats** package (5), as well as using **nlme** package for mixed-effects modeling. For more thorough coverage of the **bmbstats** package, please check the book (7).

To install these packages, please run this code:

```
install.packages(c("devtools", "tidyverse", "knitr", "lme4"), dependencies = TRUE)

devtools::install_github("mladenjovanovic/bmbstats")
```

The design of the study is two-fold (see figure below): (1) concurrent validity between 1000m time trial (1000TT) and YoYo intermittent field test level 1 (YoYOIR1) tests, and (2) repeatability of the 1000TT.

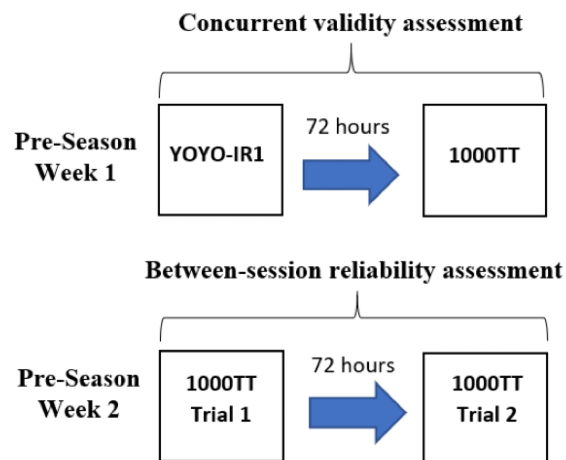


Fig. 1. Schematic of experimental timeline for assessing concurrent validity of a 1000m time trial (1000TT) compared with the Yo-Yo Intermittent Recovery Test 1 (YOYO-IR1) and between-session reliability of the 1000TT.

Dataset was provided by the author and can be found in the table below.

```
require(bmbstats)
require(tidyverse)
require(knitr)
```

```
# Load data
tt_data <- read_csv("1000TT-yoyo.csv")

kable(tt_data)
```

athlete	setup	test	value
Player 1	Validity	YOYO-IR1	2400
Player 2	Validity	YOYO-IR1	2560
Player 3	Validity	YOYO-IR1	2080
Player 4	Validity	YOYO-IR1	2440
Player 5	Validity	YOYO-IR1	2040
Player 6	Validity	YOYO-IR1	2120
Player 7	Validity	YOYO-IR1	2200
Player 8	Validity	YOYO-IR1	2280
Player 9	Validity	YOYO-IR1	2400
Player 10	Validity	YOYO-IR1	2120
Player 11	Validity	YOYO-IR1	1800
Player 12	Validity	YOYO-IR1	1520
Player 13	Validity	YOYO-IR1	1840
Player 1	Validity	1000TT	208
Player 2	Validity	1000TT	186
Player 3	Validity	1000TT	213
Player 4	Validity	1000TT	196
Player 5	Validity	1000TT	206
Player 6	Validity	1000TT	209
Player 7	Validity	1000TT	217
Player 8	Validity	1000TT	212
Player 9	Validity	1000TT	205
Player 10	Validity	1000TT	209
Player 11	Validity	1000TT	213
Player 12	Validity	1000TT	230
Player 13	Validity	1000TT	222
Player 1	Repeatability	1000TT.Trial 1	210
Player 2	Repeatability	1000TT.Trial 1	200
Player 3	Repeatability	1000TT.Trial 1	209
Player 4	Repeatability	1000TT.Trial 1	205
Player 5	Repeatability	1000TT.Trial 1	212
Player 6	Repeatability	1000TT.Trial 1	212
Player 7	Repeatability	1000TT.Trial 1	215
Player 8	Repeatability	1000TT.Trial 1	211
Player 9	Repeatability	1000TT.Trial 1	215
Player 10	Repeatability	1000TT.Trial 1	216
Player 1	Repeatability	1000TT.Trial 2	217
Player 2	Repeatability	1000TT.Trial 2	196
Player 3	Repeatability	1000TT.Trial 2	210
Player 4	Repeatability	1000TT.Trial 2	206
Player 5	Repeatability	1000TT.Trial 2	209
Player 6	Repeatability	1000TT.Trial 2	209
Player 7	Repeatability	1000TT.Trial 2	213
Player 8	Repeatability	1000TT.Trial 2	212
Player 9	Repeatability	1000TT.Trial 2	222
Player 10	Repeatability	1000TT.Trial 2	213

Predictive validity

Concurrent validity estimates the agreement between 1000TT and YoYoIR1. Rather than estimating this agreement using Pearson's r or R^2 , I would be interested in *predictive validity*. Predictive validity checks if we can predict a variable of interest (in this case YoYoIR1) within the *practical limits*. Practical limits, or *smallest-effect size of interest* (SESOI) in this case can be defined as measurement error in the variable of interest, or something that coach can find practically meaningful. Measurement error can be estimated using reliability study, or in other words random error of the test of interest (i.e., standard error of measurement; SEM, or typical error; TE). Practically meaningful SESOI can also be defined by coach, using for example difference between high and low athlete, or anything else practically meaningful. In the case these are missing, we can also use Cohen's trivial range of $0.2 \times SD$.

Here, I have defined YoYoIR1 SESOI as an average of SEMs and TEs from reliability studies (3,4) multiplied by ± 1.96 to get 95% limits of agreement.

```
YoYoIR1_SESOI <- 1.96 * mean(
  c(154, 171, 123, 137, 101, 107, 149, 77, 74, 147, 126, 172)
)
```

The selected YoYoIR1 SESOI is equal to 251m. This implies that without any *real change* in YoYoIR1 score, the test result will vary 95% within ± 251 m. If our predicted YoYoIR1 scores are within this level of agreement, our predictions are practically good and useful.

As explained previously, another way would be to use $0.2 \times SD$, which would be equal to 59m. It is up to us to define SESOI apriori to what we deem important. Here I have utilized random measurement error multiplied by 1.96.

YoYoIR1 is expressed in meters covered (i.e., distance), while 1000TT is expressed in seconds, but we can also express it as average velocity (MAS - maximum aerobic speed; in km/h). I have also utilized *adjusted mean velocity* by deducting 0.7s per each turn in 1000mTT (9 changes of direction; COD) (6).

Here is the dataset we are going to use for predictive validity analysis.

```
tt_validity <- tt_data %>%
  filter(setup == "Validity") %>%
  pivot_wider(id_cols = 1:2, names_from = "test", values_from = "value") %>%
  mutate(
    `1000TT.MAS` = 3.6 * 1000 / `1000TT`,
    `1000TT.MAS adj` = 3.6 * 1000 / (`1000TT` - 9 * 0.7)
  )

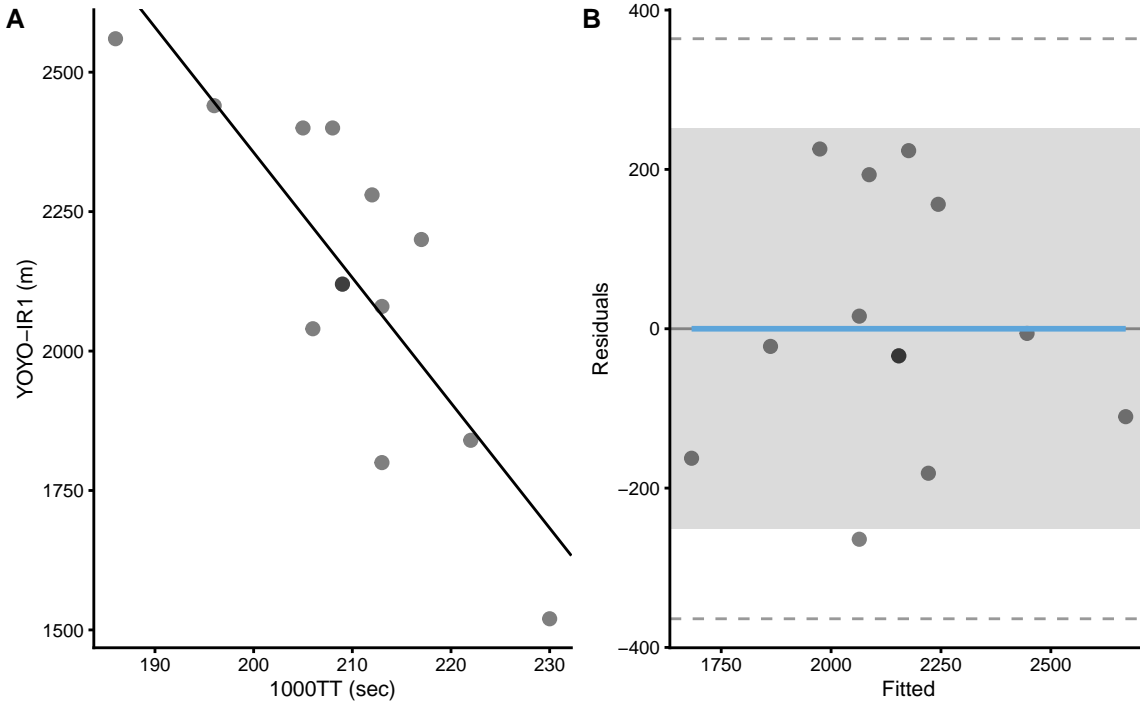
kable(tt_validity)
```

athlete	setup	YOYO-IR1	1000TT	1000TT.MAS	1000TT.MAS adj
Player 1	Validity	2400	208	17.3	17.8
Player 2	Validity	2560	186	19.4	20.0
Player 3	Validity	2080	213	16.9	17.4
Player 4	Validity	2440	196	18.4	19.0
Player 5	Validity	2040	206	17.5	18.0
Player 6	Validity	2120	209	17.2	17.8
Player 7	Validity	2200	217	16.6	17.1
Player 8	Validity	2280	212	17.0	17.5
Player 9	Validity	2400	205	17.6	18.1
Player 10	Validity	2120	209	17.2	17.8

athlete	setup	YOYO-IR1	1000TT	1000TT.MAS	1000TT.MAS adj
Player 11	Validity	1800	213	16.9	17.4
Player 12	Validity	1520	230	15.7	16.1
Player 13	Validity	1840	222	16.2	16.7

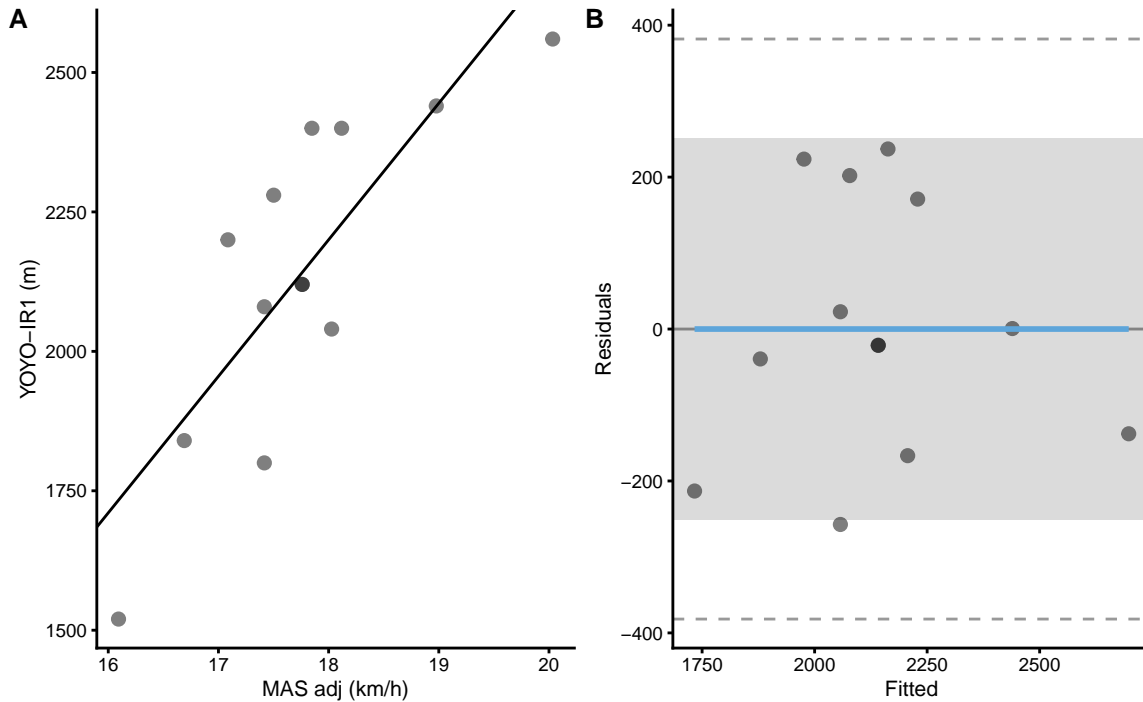
Here is scatter plot of YoYo1R and 1000mTT with residuals. Gray band indicates our selected SESOI. Dashed lines indicate SDC, or smallest detectable change, using 95% level of confidence. If residuals fall within SESOI band, then our predictions are practically useful.

```
plot_pair_lm(
  predictor = tt_validity$`1000TT`,
  predictor_label = "1000TT (sec)",
  outcome = tt_validity$`YOYO-IR1`,
  outcome_label = "YOYO-IR1 (m)",
  SESOI_lower = -YoYoIR1_SESOI,
  SESOI_upper = YoYoIR1_SESOI
)
```



Rather than using 1000TT time, we can use adjusted MAS:

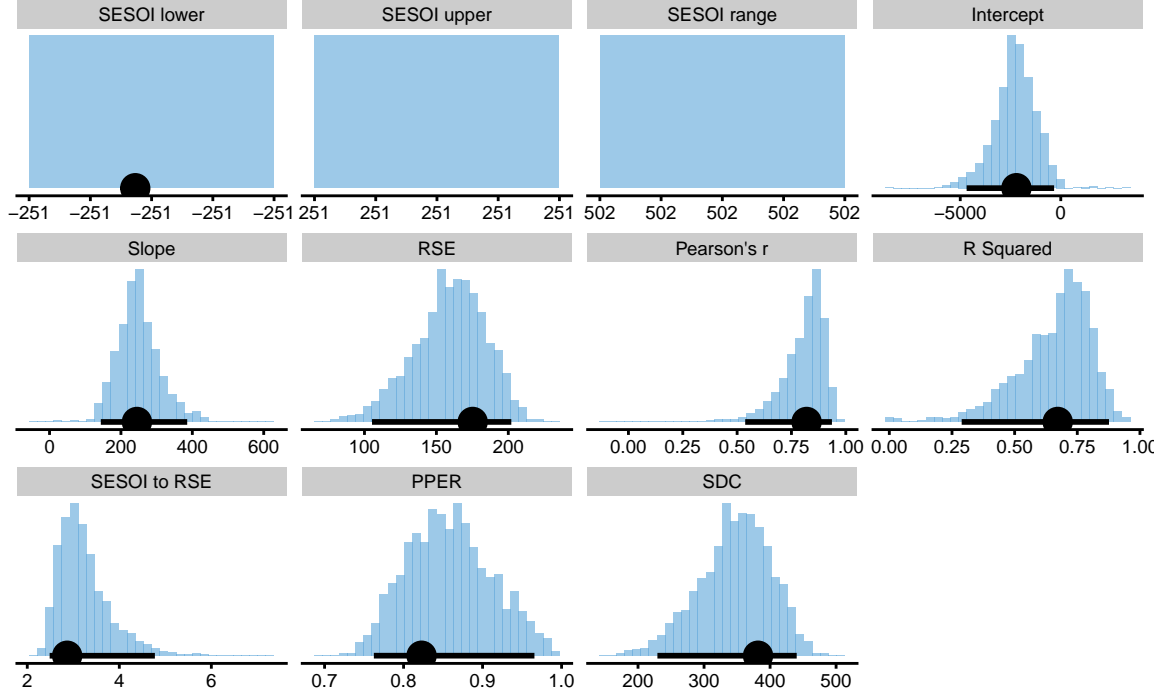
```
plot_pair_lm(
  predictor = tt_validity$`1000TT.MAS adj`,
  predictor_label = "MAS adj (km/h)",
  outcome = tt_validity$`YOYO-IR1`,
  outcome_label = "YOYO-IR1 (m)",
  SESOI_lower = -YoYoIR1_SESOI,
  SESOI_upper = YoYoIR1_SESOI
)
```



bmbstats package comes with `validity_analysis` function that provides already built in bootstrapped validity estimators (more more info see (7)).

Here I will utilize adjusted MAS to predict YoYoIR1 test distance:

```
pred_validity <- validity_analysis(
  data = tt_validity,
  criterion = "YOYO-IR1",
  practical = "1000TT.MAS adj",
  SESOI_lower = -YoYoIR1_SESOI,
  SESOI_upper = YoYoIR1_SESOI,
  control = model_control(
    boot_type = "perc",
    boot_samples = 2000,
    seed = 1667
  )
)
#> [1] "All values of t are equal to 251.206666666667 \n Cannot calculate confidence intervals"
#> [1] "All values of t are equal to 502.413333333333 \n Cannot calculate confidence intervals"
plot(pred_validity)
```



Estimators and their 95% bootstrap confidence intervals are in the table below:

```
kable(pred_validity$estimators)
```

estimator	value	lower	upper
SESOI lower	-251.207	-251.207	-251.207
SESOI upper	251.207	NA	NA
SESOI range	502.413	NA	NA
Intercept	-2206.465	-4683.758	-331.774
Slope	244.809	143.675	385.682
RSE	175.225	105.260	202.094
Pearson's r	0.820	0.537	0.936
R Squared	0.672	0.289	0.876
SESOI to RSE	2.867	2.486	4.773
PPER	0.823	0.762	0.966
SDC	381.783	229.342	440.324

PPER estimator, or proportion of practically equivalent residuals, estimates proportion of residuals within the SESOI band. This is done either by *brute force* by using counting, or assuming normal distribution of the residuals and using RSE to estimate proportion.

SDC estimator represents smallest detectable change we can detect in YoYoIR1 with 95% confidence. This is calculated using RSE and t distribution critical value (over 1.96 for small samples). If SDC is over something we deem practically useful (i.e. by using SESOI, or checking SESOI to RSE estimator), then our prediction model is not that great (practically).

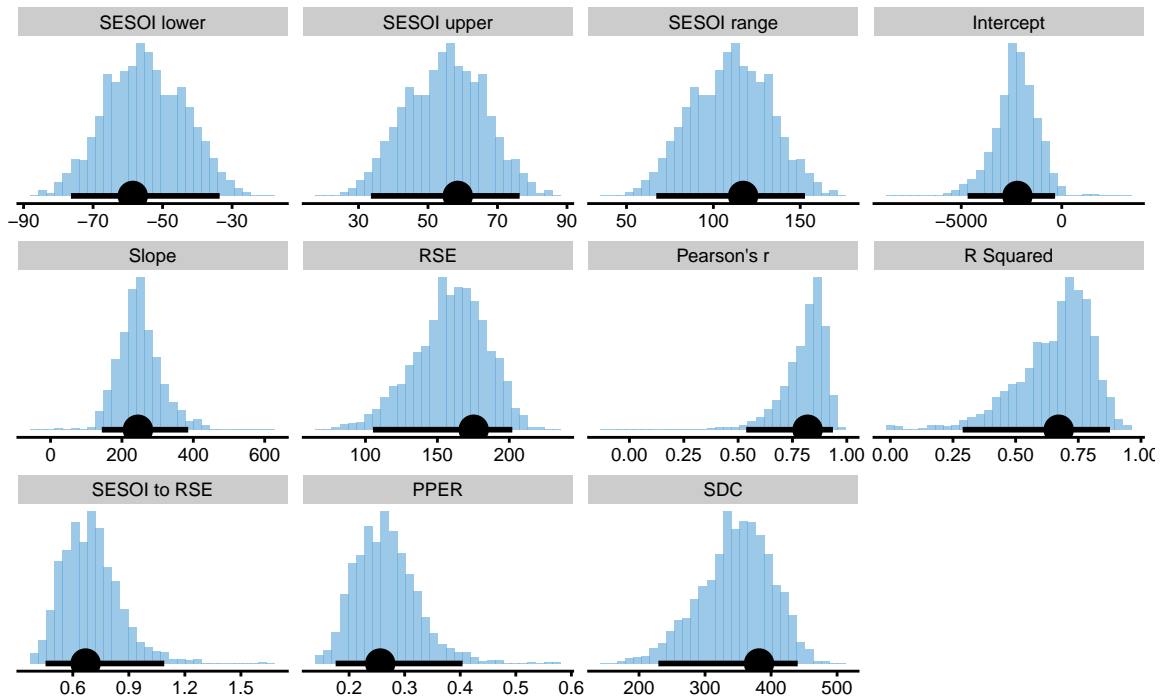
Conclusion? If we use SESOI to be $1.96 \times TE$, or 251m, then our prediction is withing practically acceptable limits (see PPER). On the other hand, if we use $0.2 \times SD$, or 59m, we can get different answer. Here is the result of that approach:

```

pred_validity <- validity_analysis(
  data = tt_validity,
  criterion = "YOYO-IR1",
  practical = "1000TT.MAS adj",
  control = model_control(
    boot_type = "perc",
    boot_samples = 2000,
    seed = 1667
  )
)

plot(pred_validity)

```



`validity_analysis` function estimates SESOI using $0.2 \times SD$ of the criterion within the bootstrap loop. That is why we have bootstrap sample for the SESOI (since it is estimated from the data, rather than provided). Estimators and their 95% bootstrap confidence intervals are in the table below:

```
kable(pred_validity$estimators)
```

estimator	value	lower	upper
SESOI lower	-58.571	-76.374	-33.568
SESOI upper	58.571	33.568	76.374
SESOI range	117.142	67.137	152.747
Intercept	-2206.465	-4683.758	-331.774
Slope	244.809	143.675	385.682
RSE	175.225	105.260	202.094
Pearson's r	0.820	0.537	0.936
R Squared	0.672	0.289	0.876

estimator	value	lower	upper
SESOI to RSE	0.669	0.454	1.090
PPER	0.256	0.176	0.404
SDC	381.783	229.342	440.324

Using this lower SESOI, brings us to different conclusion - that 1000TT is not predicting YoYoIR1 with practical agreement.

We can also evaluate this prediction using unseen data with cross-validation (CV) using `cv_model` function:

```
pred_model <- cv_model(
  `YOYO-IR1` ~ `1000TT.MAS adj`,
  tt_validity,
  control = model_control(
    cv_repeats = 10,
    cv_folds = 3
  )
)

kable(pred_model$cross_validation$performance$summary$overall)
```

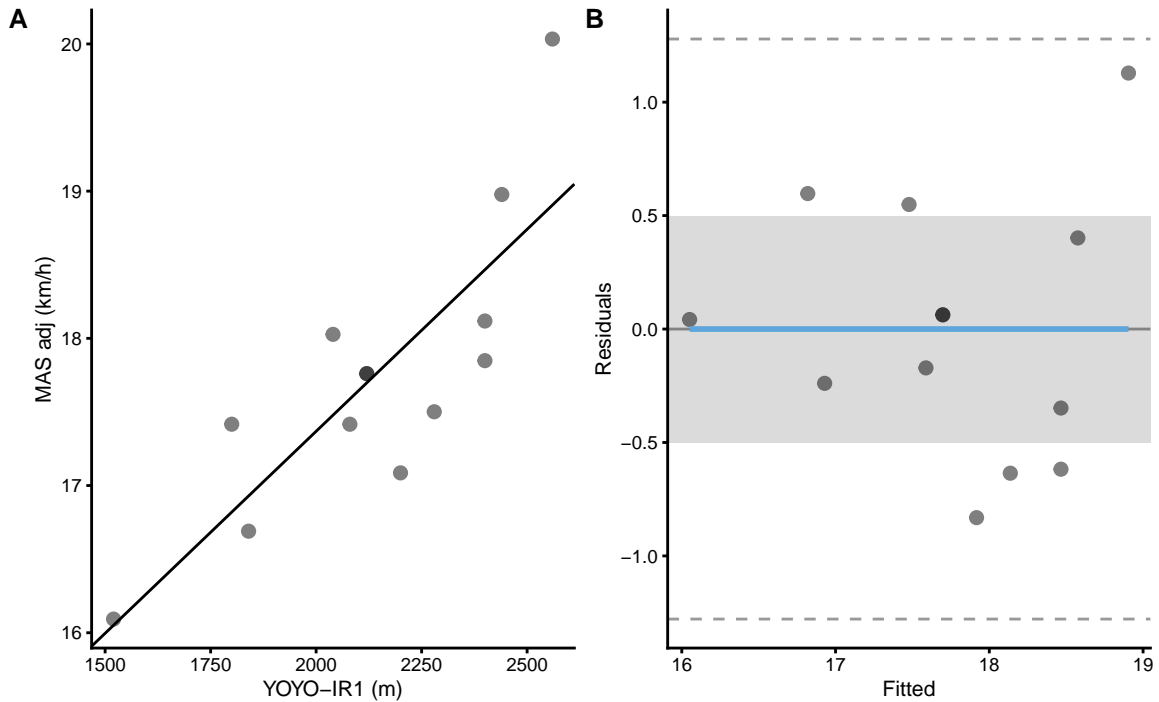
metric	training	training.pooled	testing.pooled	mean	SD	min	max
MBE	0.000	0.000	13.214	9.757	100.648	-199.308	177.227
MAE	131.881	129.499	163.029	161.951	41.530	80.898	247.702
RMSE	161.184	154.234	197.817	193.905	35.401	120.830	255.991
PPER	0.267	0.284	0.223	0.192	0.065	0.107	0.345
SESOI to RMSE	0.727	0.731	0.570	0.635	0.202	0.333	1.140
R-squared	0.672	0.700	0.508	0.504	0.241	-0.048	0.852
MinErr	-237.049	-278.159	-313.915	-214.858	64.797	-313.915	3.922
MaxErr	257.256	303.004	393.669	224.115	132.135	-55.287	393.669
MaxAbsErr	257.256	303.004	393.669	288.129	43.868	170.560	393.669

SESOI is again estimated within the CV loop using $0.2 \times SD$ of the YoYoIR1. As can be seen in the table, PPER estimated using CV is even worse (as expected). What if we use polynomial fit?

But what if we flip the task? Can Yo-YoIR1 predict 1000TT? SESOI for MAS is selected to be 0.5km/h, since that is usually the threshold utilized when grouping athletes with similar MAS scores for high-intensity interval conditioning (HIIT).

Here is the plot:

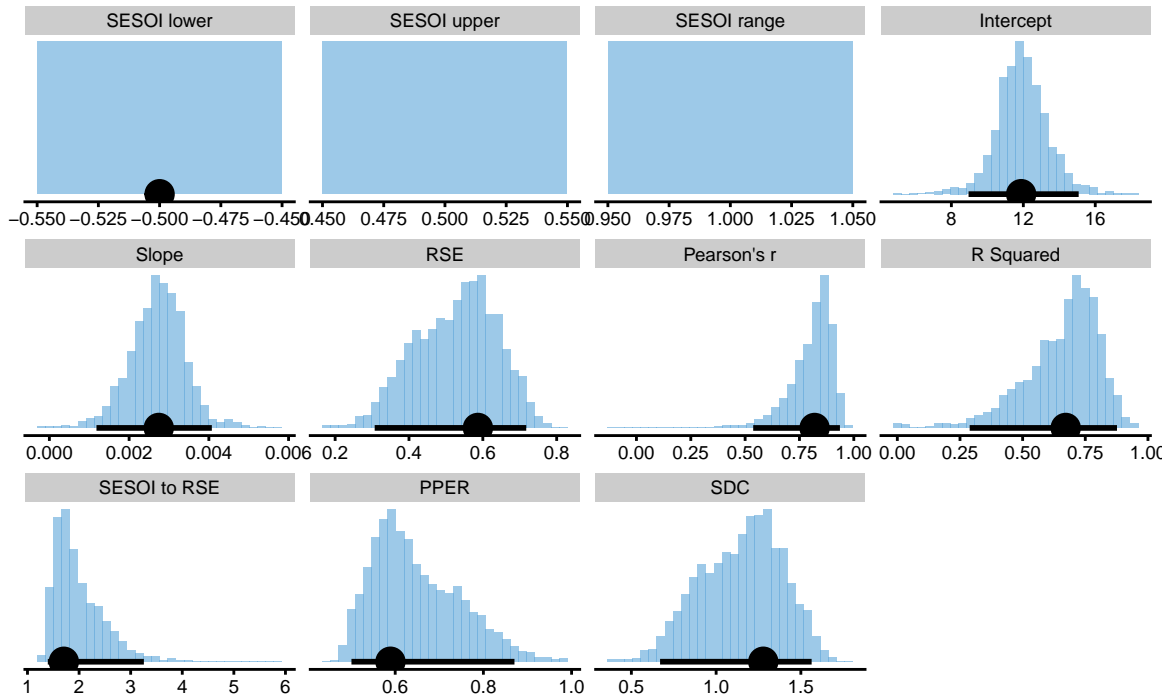
```
plot_pair_lm(
  predictor = tt_validity$`YOYO-IR1`,
  predictor_label = "YOYO-IR1 (m)",
  outcome = tt_validity$`1000TT.MAS adj`,
  outcome_label = "MAS adj (km/h)",
  SESOI_lower = -0.5,
  SESOI_upper = 0.5
)
```



And the bootstrapped estimators:

```
pred_validity <- validity_analysis(
  data = tt_validity,
  practical = "YOYO-IR1",
  criterion = "1000TT.MAS adj",
  SESOI_lower = -0.5,
  SESOI_upper = 0.5,
  control = model_control(
    boot_type = "perc",
    boot_samples = 2000,
    seed = 1667
  )
)
#> [1] "All values of t are equal to 0.5 \n Cannot calculate confidence intervals"
#> [1] "All values of t are equal to 1 \n Cannot calculate confidence intervals"

plot(pred_validity)
```



And the table:

```
kable(pred_validity$estimators)
```

estimator	value	lower	upper
SESOI lower	-0.500	-0.500	-0.500
SESOI upper	0.500	NA	NA
SESOI range	1.000	NA	NA
Intercept	11.880	8.955	15.077
Slope	0.003	0.001	0.004
RSE	0.587	0.307	0.718
Pearson's r	0.820	0.537	0.936
R Squared	0.672	0.289	0.876
SESOI to RSE	1.705	1.394	3.256
PPER	0.589	0.501	0.870
SDC	1.278	0.669	1.564

You can not the Pearson's r being the same between two analysis, but PPER being different due to different SESOI utilized. In this case, using YoYoIR1 to predict 1000TT is a bit better than vice-versa. Let's see the prediction on unseen (i.e., cross-validated data):

```
pred_model <- cv_model(
  `1000TT.MAS adj` ~ `YOYO-IR1`,
  tt_validity,
  SESOI_lower = -0.5,
  SESOI_upper = 0.5,
  control = model_control(
    cv_repeats = 10,
```

```

    cv_folds = 3
  )
)

kable(pred_model$cross_validation$performance$summary$overall)

```

metric	training	training.pooled	testing.pooled	mean	SD	min	max
MBE	0.000	0.000	-0.034	-0.027	0.344	-0.573	0.604
MAE	0.437	0.416	0.573	0.576	0.160	0.288	1.039
RMSE	0.540	0.509	0.699	0.681	0.180	0.354	1.144
PPER	0.609	0.672	0.523	0.450	0.110	0.270	0.698
SESOI to RMSE	1.853	1.963	1.430	1.572	0.431	0.874	2.822
R-squared	0.672	0.708	0.451	-0.193	1.741	-4.921	0.750
MinErr	-1.128	-1.164	-1.843	-0.818	0.625	-1.843	0.345
MaxErr	0.831	1.030	1.077	0.693	0.205	0.250	1.077
MaxAbsErr	1.128	1.164	1.843	1.089	0.342	0.598	1.843

SIMEX

The previous analyses assume that both x-variable and y-variable are *true* values. Unfortunately, they both have measurement error involved. If we know the measurement errors for the YoYo1R1 and 1000TT, we can simulate the effect of adding extra measurement error, and the extrapolating to the scenario when there is no measurement error. This procedure is called SIMEX (7–12).

For YoYo1R1 we have already used measurement error as SESOI (multiplied by 1.96), which is equal to 128m, but for 1000TT (MAS adj) we will use 0.3km/h (see the section on repeatability). Let's see how adding extra noise affects our validity estimators. To do this, for each extra noise (i.e. from 1; current level, to 3), we will make 100 simulations.

```

simex_data <- expand_grid(
  simulation = seq(1, 100),
  noise_factor = seq(1, 3, length.out = 10),
  `YoYo1R1 ME` = YoYo1R1_SESOI / 1.96,
  `1000TT.MAS adj ME` = 0.3,
  tt_validity
)

est_wrapper <- function(data) {
  # Add noise

  data <- data %>%
    mutate(
      `YOYO-IR1` = `YOYO-IR1` +
        rnorm(n(), 0, `YoYo1R1 ME` * (noise_factor-1)),

      `1000TT.MAS adj` = `1000TT.MAS adj` +
        rnorm(n(), 0, `1000TT.MAS adj ME` * (noise_factor-1))
    )

  pred_validity <- validity_estimators(
    data = data,

```

```

    criterion = "YOYO-IR1",
    practical = "1000TT.MAS adj",
    SESOI_lower = -YoYoIR1_SESOI,
    SESOI_upper = YoYoIR1_SESOI)

  as.data.frame(t(pred_validity))
}

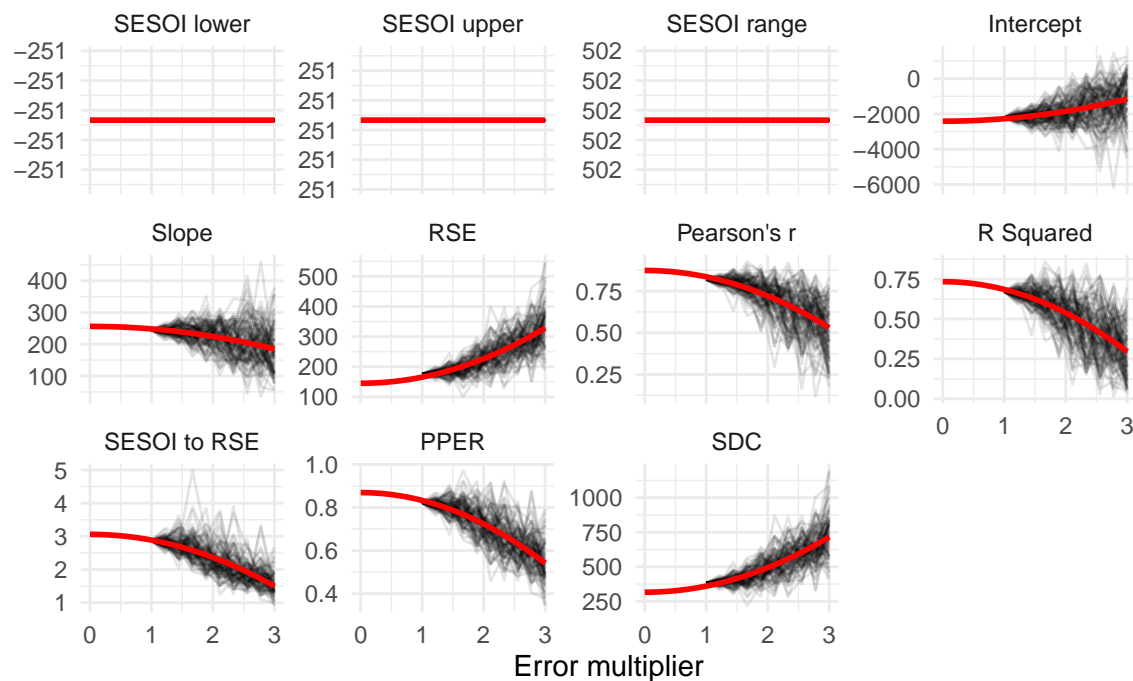
simex_res <- simex_data %>%
  group_by(simulation, noise_factor) %>%
  do(est_wrapper())

# Create SDC model
m1 <- lm(`SDC`~I(noise_factor^2), simex_res)

# Convert to long
simex_res_long <- simex_res %>%
  pivot_longer(cols = -(1:2), names_to = "estimator") %>%
  mutate(estimator = factor(estimator, levels = names(simex_res)[-(1:2)]))

# Plot
ggplot(
  simex_res_long,
  aes(x = noise_factor, group = simulation, y = value)) +
  theme_minimal() +
  geom_line(alpha = 0.1) +
  geom_smooth(
    aes(group = 1),
    method = "lm",
    formula = y ~ I(x^2),
    se = FALSE,
    color = "red",
    fullrange = TRUE
  ) +
  facet_wrap(~estimator, scales = "free_y") +
  ylab(NULL) +
  xlab("Error multiplier") +
  xlim(c(0, 3))

```



If we extrapolate to the situation when there is no measurement errors in both YoYoIR1 and 1000TT MAS adj, then SDC will be equal to 316m. Not much better, but at least we have showed the effects of measurement errors.

Repeatability

The second aim of the (2) study is to estimate 1000TT repeatability. Here is the dataset used. Please note that I have included 1000TT from the concurrent validity as well:

```
tt_repeatability <- tt_data %>%
  filter(setup == "Repeatability") %>%
  pivot_wider(id_cols = 1:2, names_from = "test", values_from = "value") %>%
  # Add TT from validity
  full_join(
    select(tt_validity, athlete, `1000TT`),
    by = "athlete") %>%
  rename(`1000TT.Trial 0` = `1000TT`)

kable(tt_repeatability)
```

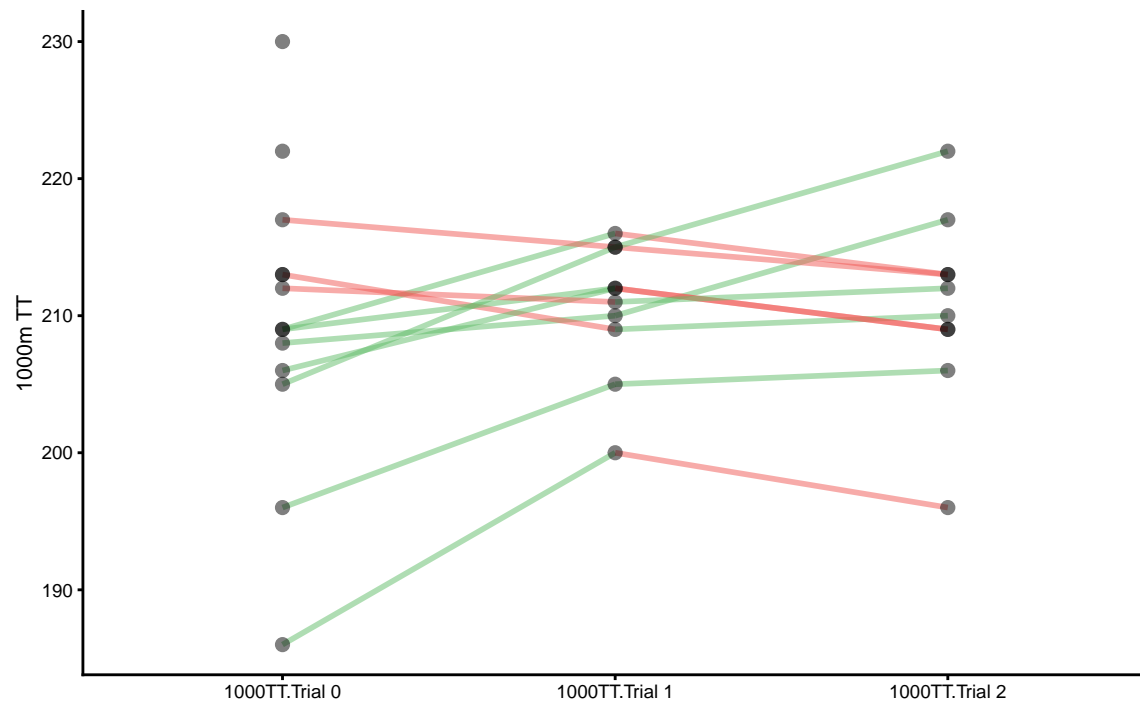
athlete	setup	1000TT.Trial 1	1000TT.Trial 2	1000TT.Trial 0
Player 1	Repeatability	210	217	208
Player 2	Repeatability	200	196	186
Player 3	Repeatability	209	210	213
Player 4	Repeatability	205	206	196
Player 5	Repeatability	212	209	206
Player 6	Repeatability	212	209	209
Player 7	Repeatability	215	213	217

athlete	setup	1000TT.Trial 1	1000TT.Trial 2	1000TT.Trial 0
Player 8	Repeatability	211	212	212
Player 9	Repeatability	215	222	205
Player 10	Repeatability	216	213	209
Player 11	NA	NA	NA	213
Player 12	NA	NA	NA	230
Player 13	NA	NA	NA	222

The following spaghetti plot depicts all three 1000TT trials (one from concurrent validity, and two from repeatability):

```
tt_repeatability_long <- pivot_longer(
  tt_repeatability,
  cols = -(1:2),
  names_to = "Trial"
) %>%
  mutate(
    Trial = factor(Trial)
  )

plot_spaghetti(
  data = tt_repeatability_long,
  id = "athlete",
  observations = "Trial",
  observations_label = NULL,
  value = "value",
  value_label = "1000m TT"
)
```

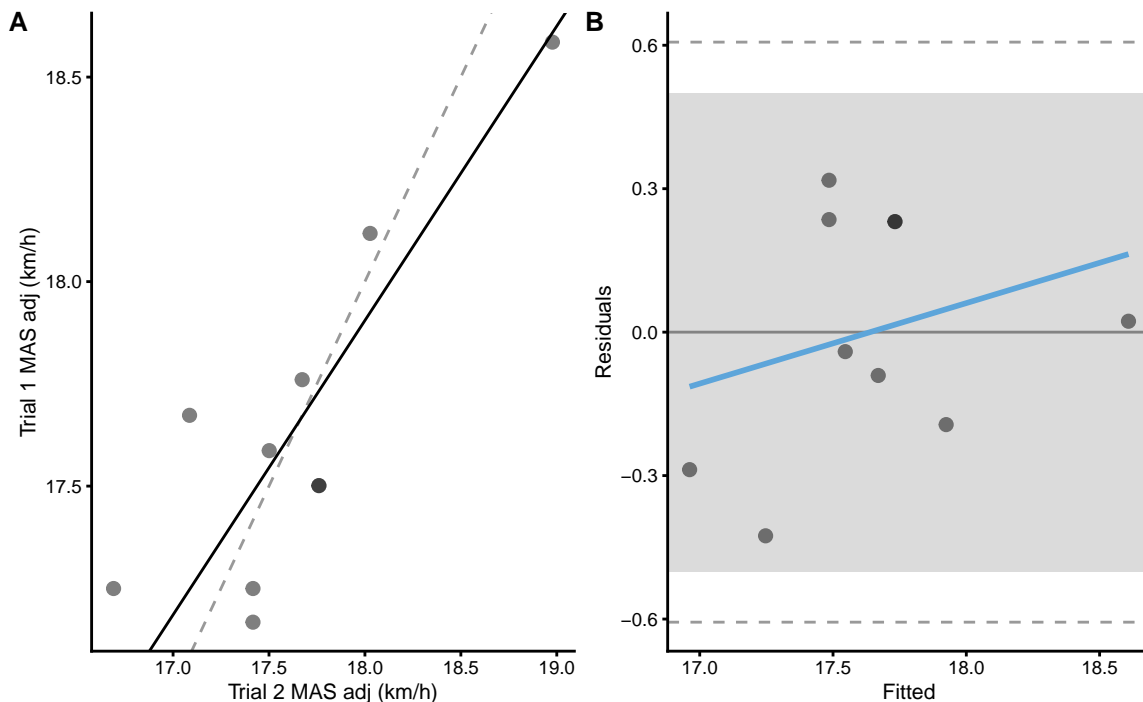


Similar to concurrent validity, we can use mean and adjusted mean velocity from the 1000TT rather than time:

```
tt_repeatability <- tt_repeatability %>%
  mutate(
    `1000TT.Trial 1.MAS` = 3.6 * 1000 / `1000TT.Trial 1`,
    `1000TT.Trial 1.MAS adj` = 3.6 * 1000 / (`1000TT.Trial 1` - 9 * 0.7),
    `1000TT.Trial 2.MAS` = 3.6 * 1000 / `1000TT.Trial 2`,
    `1000TT.Trial 2.MAS adj` = 3.6 * 1000 / (`1000TT.Trial 2` - 9 * 0.7),
    `1000TT.Trial 0.MAS` = 3.6 * 1000 / `1000TT.Trial 0`,
    `1000TT.Trial 0.MAS adj` = 3.6 * 1000 / (`1000TT.Trial 0` - 9 * 0.7)
  )
```

To estimate repeatability, only last two trial will be used. Here is the scatter plot using ordinary-least products regression (7). I am using ± 0.5 km/h as SESOI as well.

```
plot_pair_OLP(
  predictor = tt_repeatability$`1000TT.Trial 2.MAS adj`,
  predictor_label = "Trial 2 MAS adj (km/h)",
  outcome = tt_repeatability$`1000TT.Trial 1.MAS adj`,
  outcome_label = "Trial 1 MAS adj (km/h)",
  SESOI_lower = -0.5,
  SESOI_upper = 0.5,
  na.rm = TRUE
)
```



Using `reliability_analysis` we can get reliability (in this case repeatability) estimators:

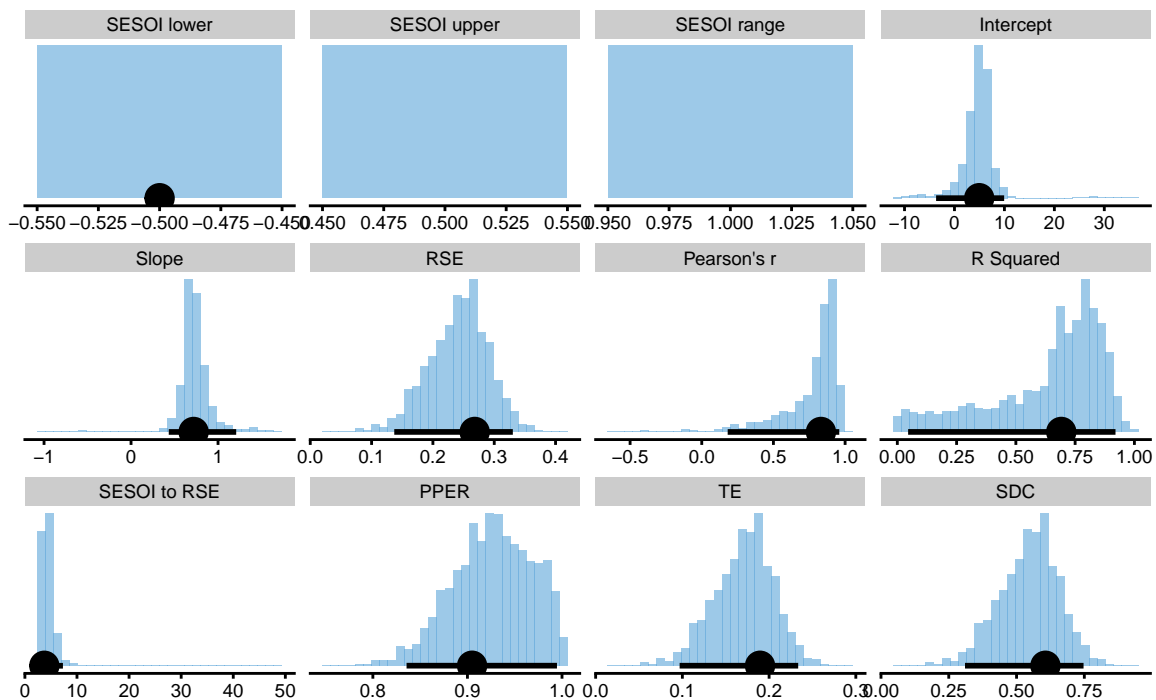
```
repeatability <- reliability_analysis(
  na.omit(tt_repeatability),
```

```

trial1 = "1000TT.Trial 1.MAS adj",
trial2 = "1000TT.Trial 2.MAS adj",
SESOI_lower = -0.5,
SESOI_upper = 0.5,
control = model_control(
  boot_type = "perc",
  boot_samples = 2000,
  seed = 1667
)
)
#> [1] "All values of t are equal to 0.5 \n Cannot calculate confidence intervals"
#> [1] "All values of t are equal to 1 \n Cannot calculate confidence intervals"

plot(repeatability)

```



And the table:

```
kable(repeatability$estimators)
```

estimator	value	lower	upper
SESOI lower	-0.500	-0.500	-0.500
SESOI upper	0.500	NA	NA
SESOI range	1.000	NA	NA
Intercept	4.951	-3.679	9.948
Slope	0.720	0.434	1.209
RSE	0.268	0.137	0.330
Pearson's r	0.832	0.181	0.959
R Squared	0.691	0.046	0.920

estimator	value	lower	upper
SESOI to RSE	3.730	3.028	7.301
PPER	0.905	0.836	0.995
TE	0.190	0.097	0.234
SDC	0.607	0.310	0.747

Here we have TE estimated, as well as SDC, which is roughly equal to $TE \times \sqrt{2} \times 1.96$. That is the smallest change in the 1000TT MAS that we have 95% confidence in that is *real*, and not biological and measurement fluke. As can be seen, that is pretty close to our SESOI of 0.5km/h. This is also indicated with excellent PPER estimator.

But what if we take into account first trial, done in the construct validity part of this study?

Mixed-effects model

Package `lme4` (1) is utilized to fit linear mixed models.

```
require(lme4)
require(stringr)

# Convert to adj MAS
tt_repeatability_long <- tt_repeatability_long %>%
  mutate(
    Trial = factor(Trial),
    trial = as.numeric(str_sub(Trial, -1)),
    value = 3.6 * 1000 / (value - 9 * 0.7)
  )

# Using trials as numeric
mm1 <- lme4::lmer(
  value ~ trial + (1 + trial | athlete),
  data = tt_repeatability_long
)

summary(mm1)
#> Linear mixed model fit by REML ['lmerMod']
#> Formula: value ~ trial + (1 + trial | athlete)
#> Data: tt_repeatability_long
#>
#> REML criterion at convergence: 56.9
#>
#> Scaled residuals:
#>    Min      1Q  Median      3Q     Max
#> -1.537 -0.616  0.201  0.464  1.432
#>
#> Random effects:
#> Groups   Name              Variance Std.Dev. Corr
#> athlete (Intercept) 0.7229    0.850
#>          trial        0.0565    0.238   -0.88
#> Residual              0.1012    0.318
#> Number of obs: 33, groups: athlete, 13
#>
```

```

#> Fixed effects:
#>               Estimate Std. Error t value
#> (Intercept)  17.7009      0.2498   70.87
#> trial        -0.1304      0.0961   -1.36
#>
#> Correlation of Fixed Effects:
#>      (Intr)
#> trial -0.736

# Using trials as ordinal
mm2 <- lme4::lmer(
  value ~ Trial + (1 | athlete),
  data = tt_repeatability_long
)

summary(mm2)
#> Linear mixed model fit by REML ['lmerMod']
#> Formula: value ~ Trial + (1 | athlete)
#> Data: tt_repeatability_long
#>
#> REML criterion at convergence: 60.1
#>
#> Scaled residuals:
#>      Min       1Q   Median       3Q      Max
#> -1.5034 -0.4653  0.0856  0.4724  1.9798
#>
#> Random effects:
#> Groups Name      Variance Std.Dev.
#> athlete (Intercept) 0.540    0.735
#> Residual          0.135    0.367
#> Number of obs: 33, groups: athlete, 13
#>
#> Fixed effects:
#>               Estimate Std. Error t value
#> (Intercept)      17.748      0.228   77.94
#> Trial1000TT.Trial 1  -0.353      0.162   -2.18
#> Trial1000TT.Trial 2  -0.361      0.162   -2.23
#>
#> Correlation of Fixed Effects:
#>      (Intr) T10001
#> Tr1000TT.T1 -0.280
#> Tr1000TT.T2 -0.280  0.488

```

Plot

```

tt_repeatability_mm1 <- tt_repeatability_long %>%
  mutate(
    pred = predict(mm1, tt_repeatability_long),
    resid = pred - value
  )

tt_repeatability_mm2 <- tt_repeatability_long %>%
  mutate(

```

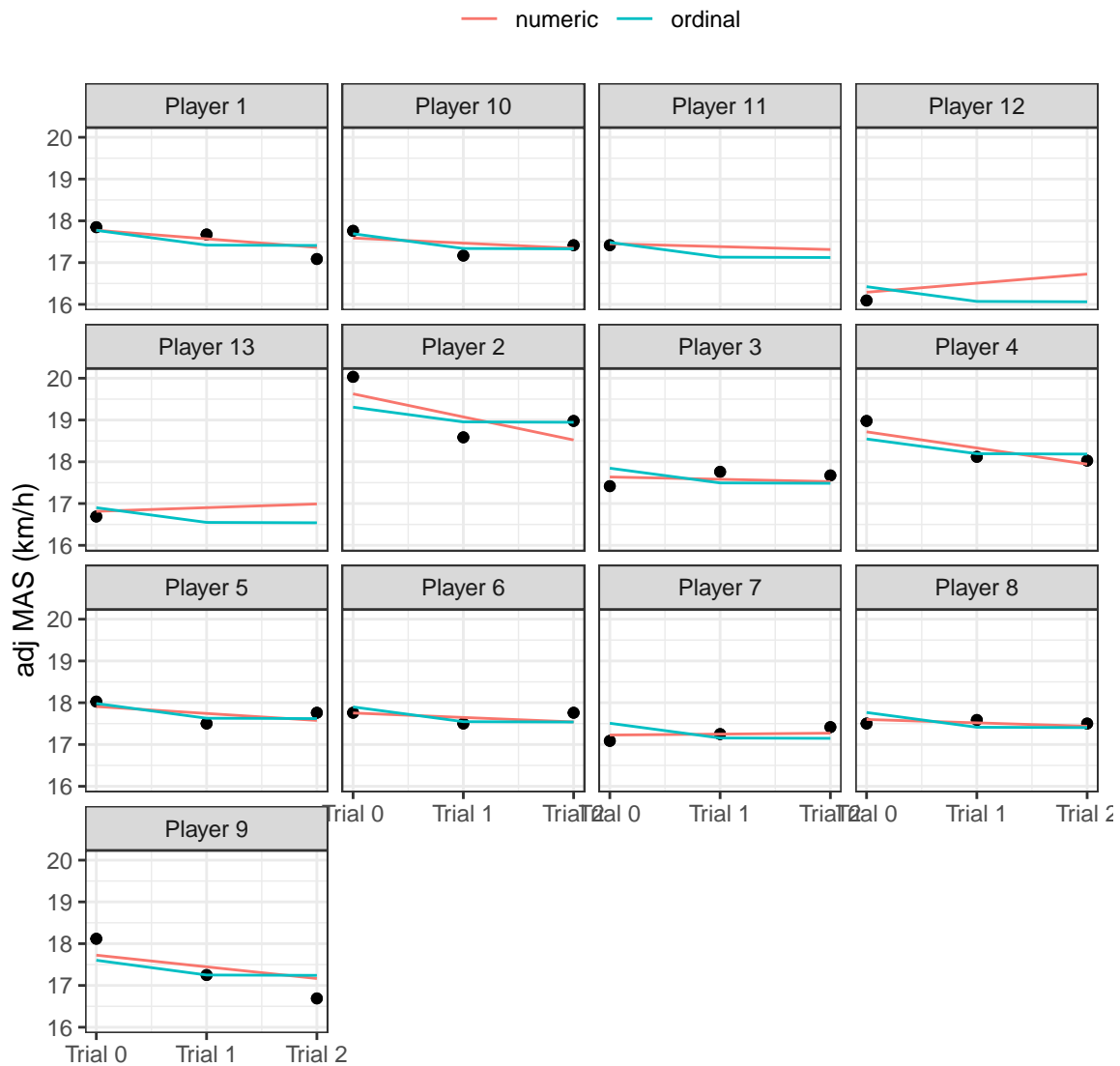
```

    pred = predict(mm2, tt_repeatability_long),
    resid = pred - value
  )

tt_repeatability_pred <- rbind(
  data.frame(tt_repeatability_mm1, model = "numeric"),
  data.frame(tt_repeatability_mm2, model = "ordinal")
)

ggplot(
  tt_repeatability_pred,
  aes(x = trial, y = value)
) +
  theme_bw() +
  geom_point() +
  geom_line(aes(y = pred, color = model, group = model)) +
  facet_wrap(~athlete) +
  scale_x_continuous(
    breaks = c(0, 1, 2),
    labels = c("Trial 0", "Trial 1", "Trial 2")) +
  xlab(NULL) +
  ylab("adj MAS (km/h)") +
  theme(
    legend.position = "top",
    legend.title = element_blank()
  )

```



Bootstrapping all methods to get typical error

Remove the three individuals with NAs.

```
boot_data <- na.omit(tt_repeatability)

te_estimators <- function(data, SESOI_lower, SESOI_upper, na.rm, init_boot) {
  te_diff <- sd(data$`1000TT.Trial 2.MAS adj` - data$`1000TT.Trial 1.MAS adj`) / sqrt(2)

  te_lm <- summary(
    lm(`1000TT.Trial 2.MAS adj` ~ `1000TT.Trial 1.MAS adj`, data)
  )$sigma / sqrt(2)

  te_olp <- OLP_regression(
    predictor = data$`1000TT.Trial 1.MAS adj`,
    outcome = data$`1000TT.Trial 2.MAS adj`
  )
}
```

```

)$rse / sqrt(2)

data_long <- pivot_longer(
  select(
    data,
    1:2,
    `1000TT.Trial 0.MAS adj`,
    `1000TT.Trial 1.MAS adj`,
    `1000TT.Trial 2.MAS adj`),
  cols = -(1:2),
  names_to = "Trial")

data_long <- data_long %>%
mutate(
  Trial = factor(Trial),
  trial = as.numeric(str_sub(Trial, -9, -9))
)

mm1 <- lme4::lmer(
  value ~ trial + (1 + trial | athlete),
  data = data_long
)

mm2 <- lme4::lmer(
  value ~ Trial + (1 | athlete),
  data = data_long
)

c(
  "diff TE" = te_diff,
  "lm TE" = te_lm,
  "olp TE" = te_olp,
  "MM numeric" = summary(mm1)$sigma,
  "MM nominal" = summary(mm2)$sigma
)
}

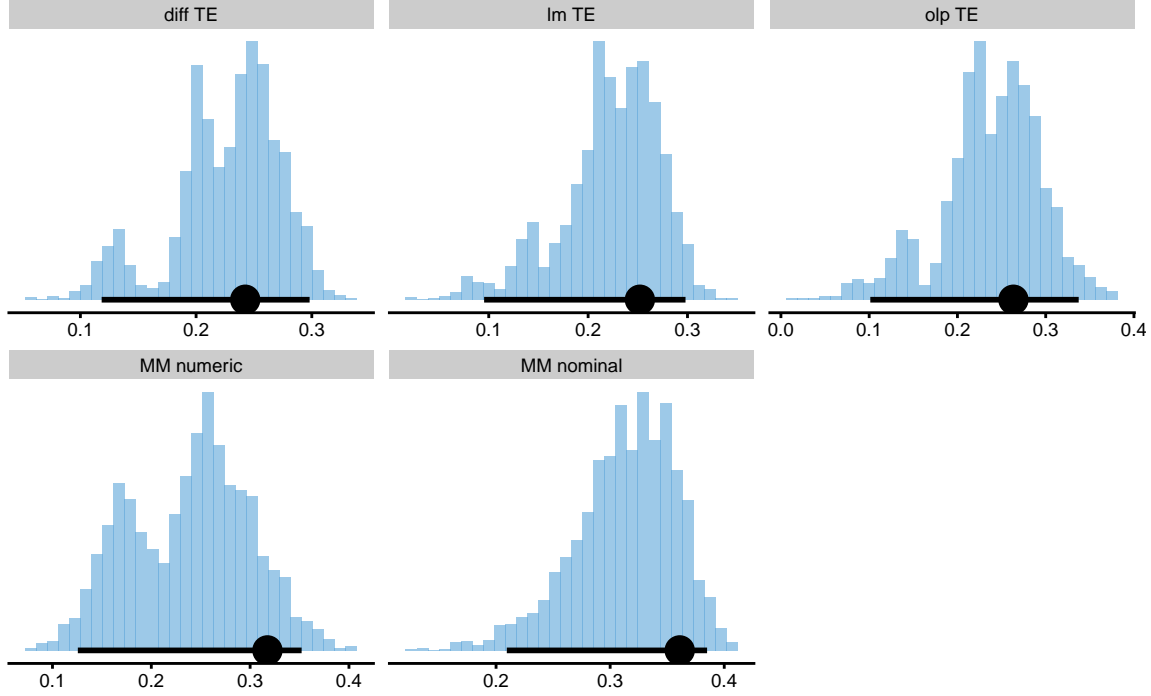
boot_te <- bmbstats(
  data = boot_data,
  estimator_function = te_estim实施ors,
  control = model_control(
    iter = FALSE,
    boot_type = "perc",
    boot_samples = 2000,
    seed = 1667
  )
)

plot(boot_te)

kable(boot_te$estimators)

```

estimator	value	lower	upper
diff TE	0.242	0.118	0.298
lm TE	0.252	0.095	0.298
olp TE	0.263	0.101	0.337
MM numeric	0.318	0.126	0.352
MM nominal	0.361	0.209	0.385



References

1. Bates, D, Mächler, M, Bolker, B, and Walker, S. Fitting linear mixed-effects models using lme4. *Journal of Statistical Software* 67: 1–48, 2015.
2. Clancy, C, Green, P, Curnyn, S, Donaldson, E, and Ring, N. The concurrent validity and between-session reliability of a 1000m time trial for the assessment of aerobic fitness in elite development soccer players. *Sport Performance & Science Reports* 3, 2020.
3. Deprez, D, Coutts, A, Lenoir, M, Fransen, J, Pion, J, Philippaerts, R, et al. Reliability and validity of the Yo-Yo intermittent recovery test level 1 in young soccer players. *Journal of sports sciences* 32, 2014.
4. Deprez, D, Fransen, J, Lenoir, M, Philippaerts, R, and Vaeyens, R. The Yo-Yo Intermittent Recovery test level 1 is reliable in young high-level soccer players. *Biology of Sport* 32, 2014.
5. Jovanović, M. bmbstats: Bootstrap magnitude-based statistics. Belgrade, Serbia, 2020. Available from: <https://github.com/mladenjovanovic/bmbstats>
6. Jovanović, M. HIIT Manual: High Intensity Interval Training and Agile Periodization. Ultimate Athlete Concepts, 2018.
7. Jovanović, M. Bmbstats: Bootstrap Magnitude-based Statistics for Sports Scientists. Mladen Jovanović, 2020.

8. Keogh, RH, Shaw, PA, Gustafson, P, Carroll, RJ, Deffner, V, Dodd, KW, et al. STRATOS guidance document on measurement error and misclassification of variables in observational epidemiology: Part 1-Basic theory and simple methods of adjustment. *Statistics in Medicine*, 2020.
9. Lederer, W and Küchenhoff, H. A short introduction to the SIMEX and MCSIMEX. *R News* 6, 2006.
10. Shang, Y. Measurement Error Adjustment Using the SIMEX Method: An Application to Student Growth Percentiles: *Measurement Error Adjustment Using the SIMEX Method. Journal of Educational Measurement* 49: 446–465, 2012.
11. Shaw, PA, Gustafson, P, Carroll, RJ, Deffner, V, Dodd, KW, Keogh, RH, et al. STRATOS guidance document on measurement error and misclassification of variables in observational epidemiology: Part 2-More complex methods of adjustment and advanced topics. *Statistics in Medicine*, 2020.
12. Wallace, M. Analysis in an imperfect world. *Significance* 17: 14–19, 2020.