Week 5 Statistics

2024-11-04

1.1 Read in Data

Perform demographics table and perform group comparisons

https://www.danieldsjoberg.com/gtsummary/articles/tbl_summary.html

```
rm(list = ls())

library(ggplot2)
library(readxl)
library(nlme)
library(gtsummary)
library(huxtable)
```

```
##
## Attaching package: 'huxtable'

## The following object is masked from 'package:ggplot2':
##
##     theme_grey
```

```
library(officer)
```

```
##
## Attaching package: 'officer'

## The following objects are masked from 'package:huxtable':
##
##     to_html, to_rtf

## The following object is masked from 'package:readxl':
##
##     read_xlsx
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:huxtable':
##
##     add_rownames

## The following object is masked from 'package:nlme':
##
##     collapse

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
demo=read_excel("SubjectInfo.xlsx")

str(demo)
```

```
## tibble [13 x 8] (S3: tbl_df/tbl/data.frame)
##  $ Subject No                  : chr [1:13] "Sub1" "Sub2" "Sub3" "Sub4" ...
##  $ Age                         : num [1:13] 26 28 21 25 34 19 21 21 25 20 ...
##  $ Reported Weight (kg)        : num [1:13] 86 77 52 73 86 54 59 57 58 66 ...
##  $ Reported Length (cm)        : num [1:13] 185 178 170 168 173 160 163 173 170 170 ...
##  $ Gender                      : chr [1:13] "M" "F" "M" "M" ...
##  $ Level Slow                  : num [1:13] 886 768 531 NA 879 ...
##  $ Level Walk                  : num [1:13] 892 760 558 NA 899 ...
##  $ Weight from force plates(kg): num [1:13] 90.6 77.9 55.5 NA 90.6 ...
```

```
demo.clean=demo[,c(-1)]
demo.clean$Age=as.numeric(demo.clean$Age)

tb=tbl_summary(demo.clean,
            statistic = list(
                 all_continuous() ~ "{mean} ({sd})",
                 all_categorical() ~ "{n} ({p}%)"
               ),
            type = list(Age ~ "continuous",`Reported Length (cm)`~"continuous"),
               digits = all_continuous() ~ 2, #I want two points of precision (two decimal points) for
               missing = "ifany") %>%
  bold_labels()

tb
```

Characteristic	**N = 13**[1]
Age	24.15 (4.96)
Reported Weight (kg)	66.92 (11.93)
Reported Length (cm)	172.69 (7.93)
Gender	

F	6 (46%)
M	7 (54%)
Level Slow	685.77 (121.67)
Unknown	1
Level Walk	690.72 (120.88)
Unknown	1
Weight from force plates(kg)	70.16 (12.35)
Unknown	1

[1] Mean (SD); n (%)

```
tb.group=tbl_summary(demo.clean,
               by = Gender,
            statistic = list(
                 all_continuous() ~ "{mean} ({sd})",
                 all_categorical() ~ "{n} ({p}%)"
               ),
            type = list(Age ~ "continuous",`Reported Length (cm)`~"continuous"),
               digits = all_continuous() ~ 2, #I want two points of precision (two decimal points) for
               missing = "ifany") %>%
  add_p(
    #test = list("Age" = "t.test")
    ) %>%
  bold_labels()
```

```
## The following warnings were returned during `add_p()`:

## ! For variable `Age` (`Gender`) and "estimate", "statistic", "p.value",
##   "conf.low", and "conf.high" statistics: cannot compute exact p-value with
##   ties
## ! For variable `Age` (`Gender`) and "estimate", "statistic", "p.value",
##   "conf.low", and "conf.high" statistics: cannot compute exact confidence
##   intervals with ties
## ! For variable `Reported Length (cm)` (`Gender`) and "estimate", "statistic",
##   "p.value", "conf.low", and "conf.high" statistics: cannot compute exact
##   p-value with ties
## ! For variable `Reported Length (cm)` (`Gender`) and "estimate", "statistic",
##   "p.value", "conf.low", and "conf.high" statistics: cannot compute exact
##   confidence intervals with ties
## ! For variable `Reported Weight (kg)` (`Gender`) and "estimate", "statistic",
##   "p.value", "conf.low", and "conf.high" statistics: cannot compute exact
##   p-value with ties
## ! For variable `Reported Weight (kg)` (`Gender`) and "estimate", "statistic",
##   "p.value", "conf.low", and "conf.high" statistics: cannot compute exact
##   confidence intervals with ties
```

```
tb.group
```

Characteristic	**F**

N = 6[1] N = 7[1]	**M** **p-value**[2]		
Age	21.67 (3.20)	26.29 (5.41)	0.13
Reported Weight (kg)	61.50 (8.64)	71.57 (12.96)	0.2
Reported Length (cm)	168.83 (6.55)	176.00 (7.90)	0.2
Level Slow	643.42 (75.09)	728.12 (150.42)	0.4
Unknown	0	1	
Level Walk	642.33 (71.74)	739.11 (146.23)	0.4
Unknown	0	1	
Weight from force plates(kg)	65.53 (7.48)	74.78 (15.11)	0.4
Unknown	0	1	

[1] Mean (SD)
[2] Wilcoxon rank sum test; Wilcoxon rank sum exact test

```
table1=as_hux_table(tb.group)
#Convert to word doc which will be saved in your current path
quick_docx(table1,file="Gender Demo table.docx")
```

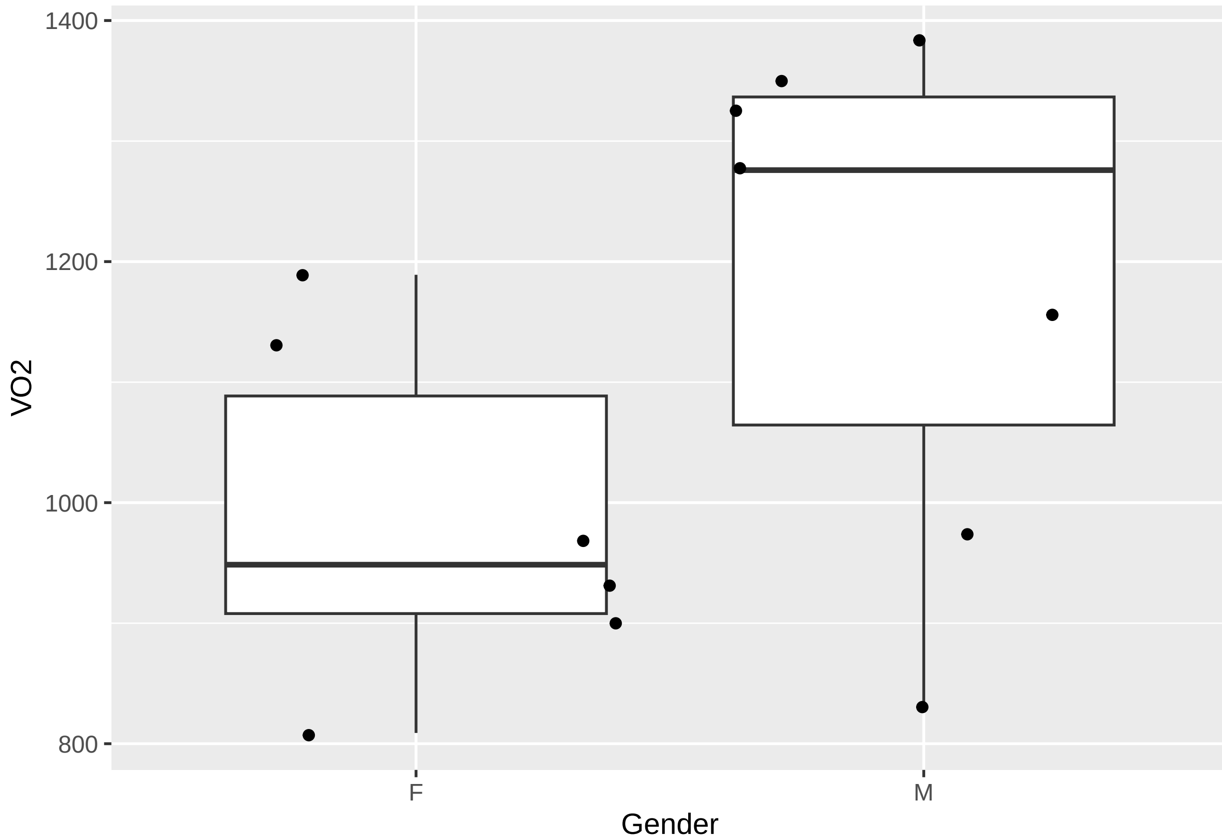
```
## Registered S3 method overwritten by 'ftExtra':
##   method                  from
##   as_flextable.data.frame flextable
```

1.2 Linear Models

```
data.task=read.csv("raw.data.all.csv")
data.all=merge(data.task,demo,by.x = "Sub",by.y = "Subject No")

#Aggregate all data by subject age and gender
data.agg=aggregate(VO2~Age+Gender+Sub,data.all,mean)

aggregate(VO2~Gender,data.all,mean)
```

Gender	**VO2**
F	985
M	1.19e+03

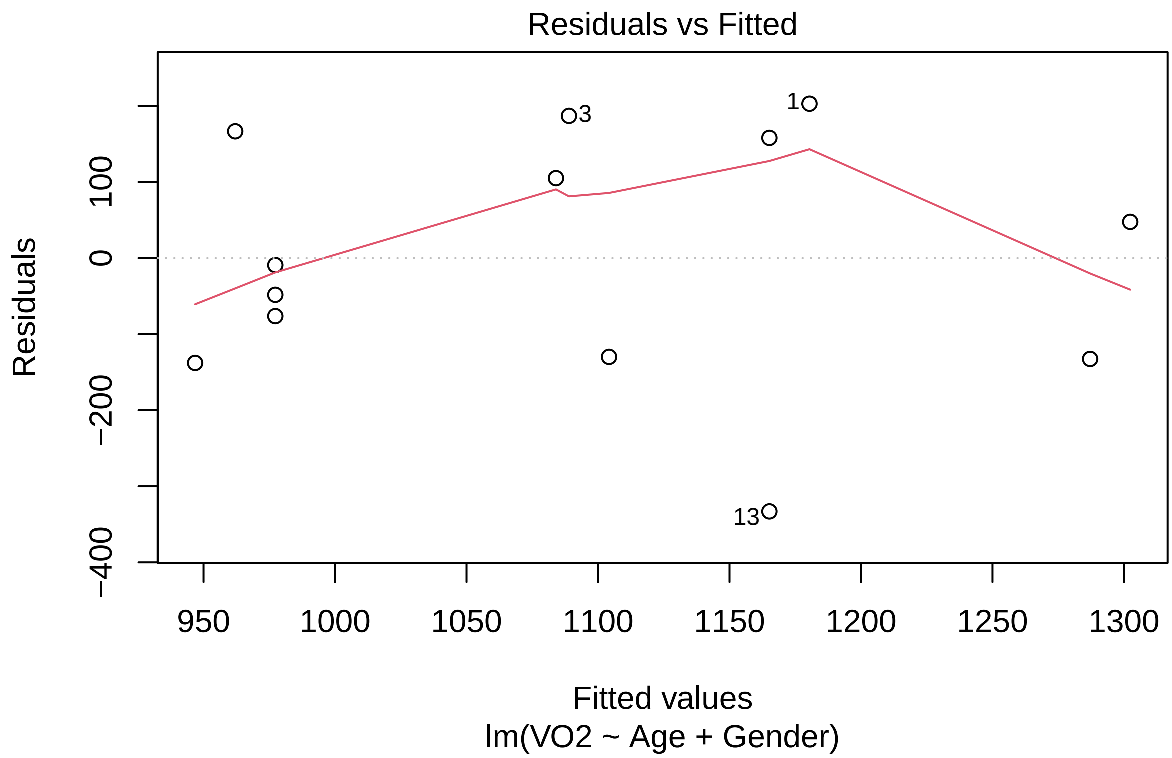
```
ggplot(data.agg,aes(x=Gender,y=VO2))+
  geom_boxplot()+
  geom_jitter()
```

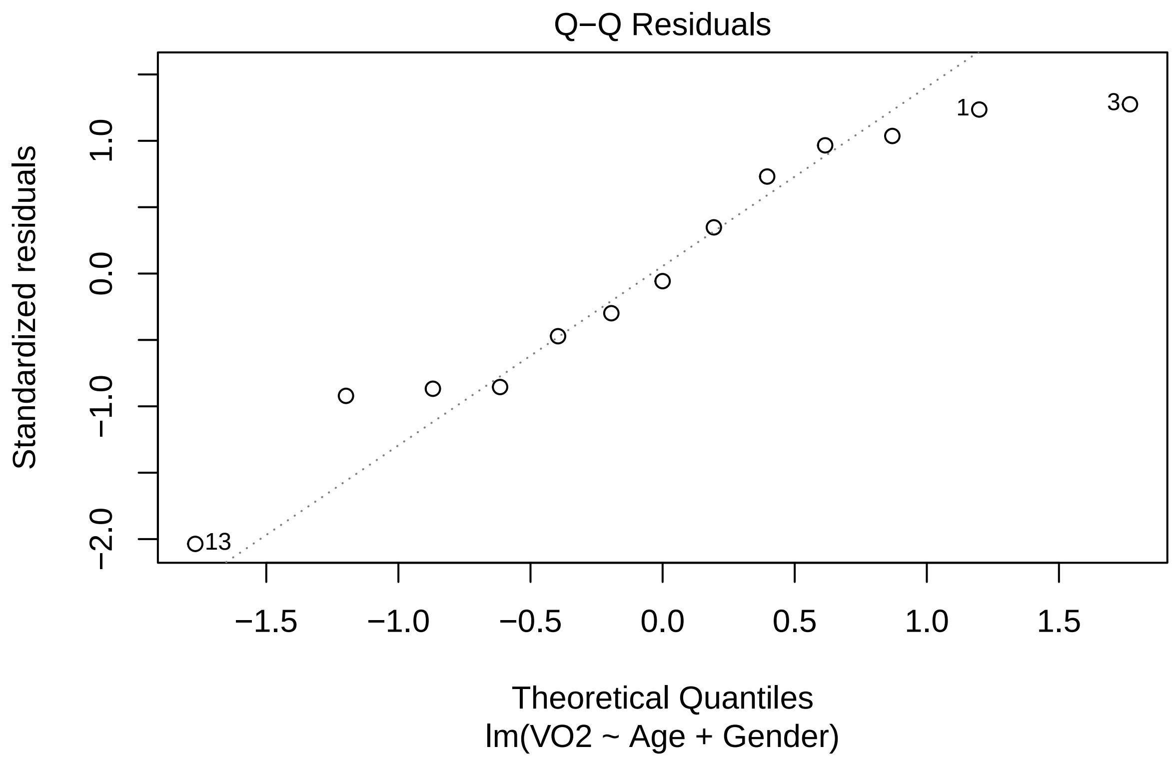



```
library(car)
```

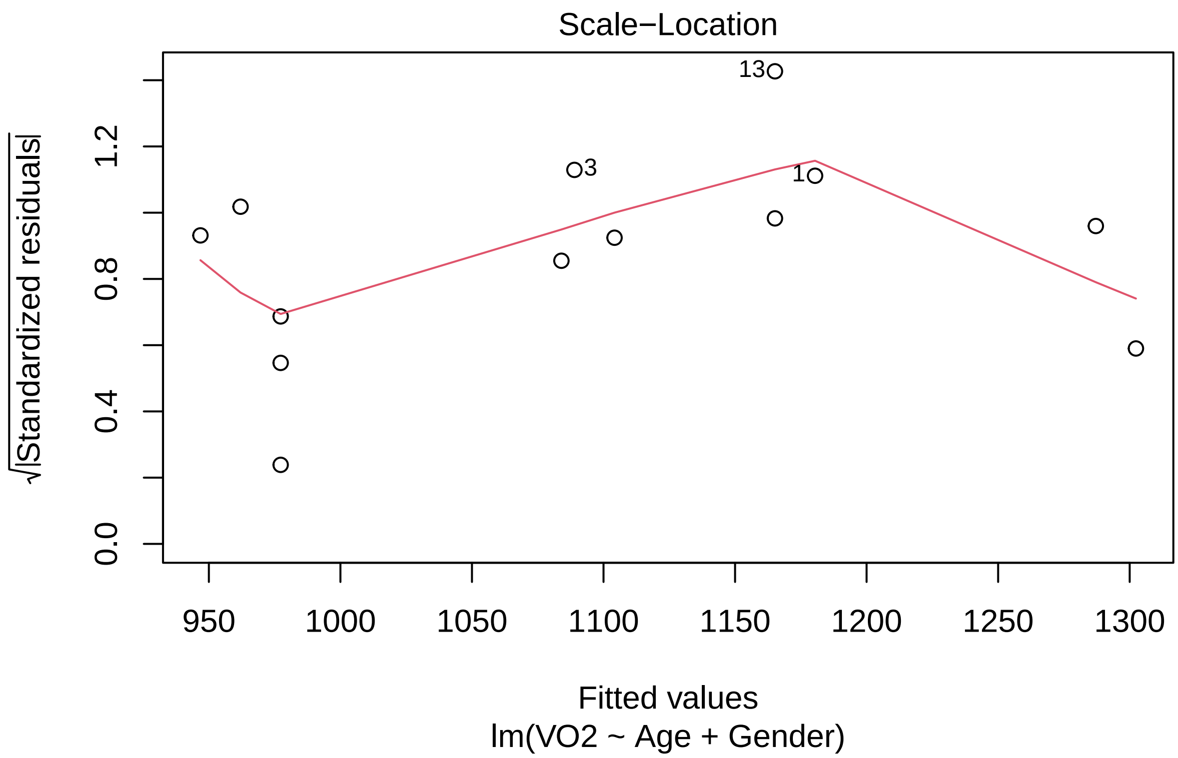
```
## Loading required package: carData

##
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':
##
##     recode
```

```
data.agg$Gender=factor(data.agg$Gender,levels = c("F","M"))
#Run first linear model with lm()
fit=lm(VO2~Age+Gender,data.agg)
#check assumptions
plot(fit)
```

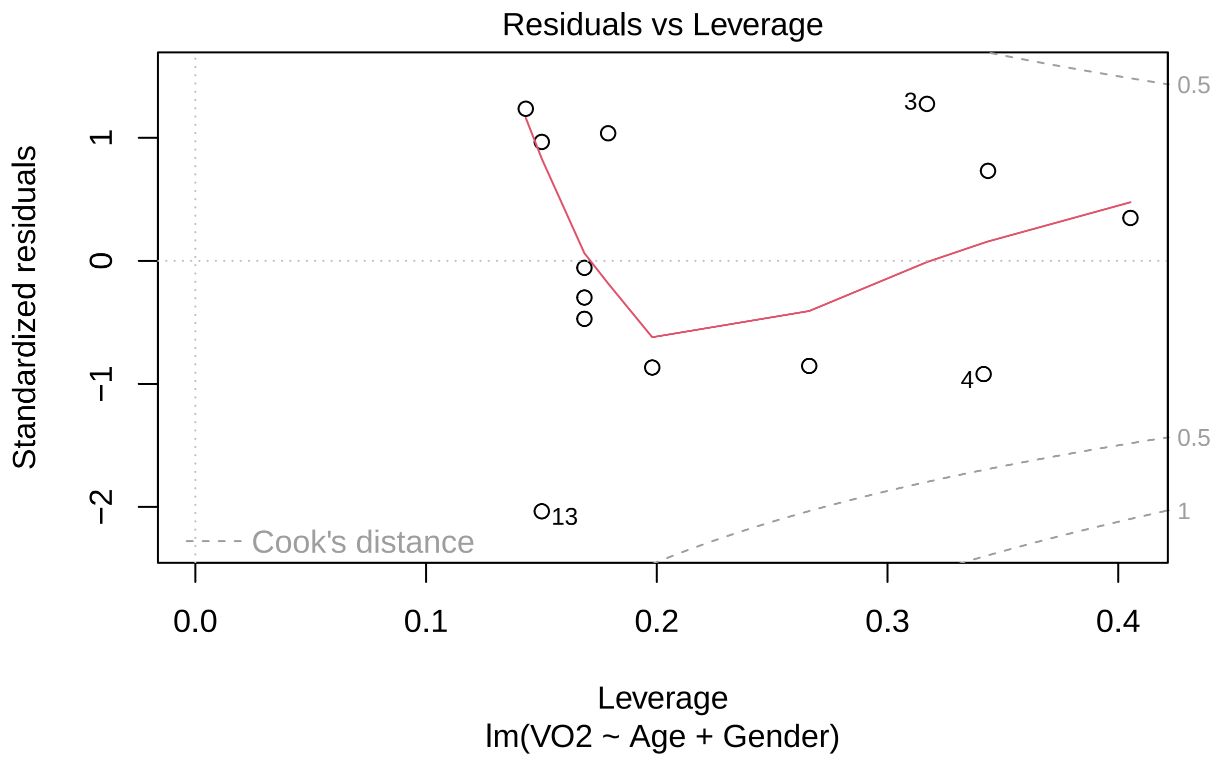

Residuals vs Fitted
Residuals
Fitted values
lm(VO2 ~ Age + Gender)
100
0
−200
−400
950
1000
1050
1100
1150
1200
1250
1300
3
1
13



Q−Q Residuals
Standardized residuals
Theoretical Quantiles
lm(VO2 ~ Age + Gender)
−2.0
−1.0
0.0
1.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
13
1
3



Scale−Location
√|Standardized residuals|
0.0
0.4
0.8
1.2
13
3
1
950
1000
1050
1100
1150
1200
1250
1300
Fitted values
lm(VO2 ~ Age + Gender)




```r
vif(fit) #Check for collinearity
```

```
##      Age   Gender
## 1.303977 1.303977
```

```r
cooks.distance(fit) #Check for outliers
```

```
##           1           2           3           4           5           6
## 0.085061356 0.078027542 0.251698070 0.146790987 0.015047893 0.093197866
##           7           8           9          10          11          12
## 0.088249850 0.054965527 0.027529781 0.061953889 0.006031350 0.000219273
##          13
## 0.244149178
```

```r
summary(fit)
```

```
##
## Call:
## lm(formula = VO2 ~ Age + Gender, data = data.agg)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -333.01 -129.89   -9.21  158.01  202.90
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   657.13     265.34   2.477   0.0327 *
## Age            15.25      11.78   1.294   0.2248
## GenderM       126.91     112.71   1.126   0.2865
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 177.4 on 10 degrees of freedom
## Multiple R-squared:  0.3619, Adjusted R-squared:  0.2343
## F-statistic: 2.836 on 2 and 10 DF,  p-value: 0.1058
```

```
confint(fit)
```

```
##                  2.5 %     97.5 %
## (Intercept)   65.92321 1248.34618
## Age          -11.00538   41.49554
## GenderM     -124.22584  378.04038
```

```
tbl_regression(fit,
               show_single_row = "Gender",
               label = list(Gender = "Gender(Male)"),
               intercept = T)
```

Characteristic	Beta	95% CI[1]	p-value
(Intercept)	657	66, 1,248	0.033
Age	15	-11, 41	0.2
Gender(Male)	127	-124, 378	0.3

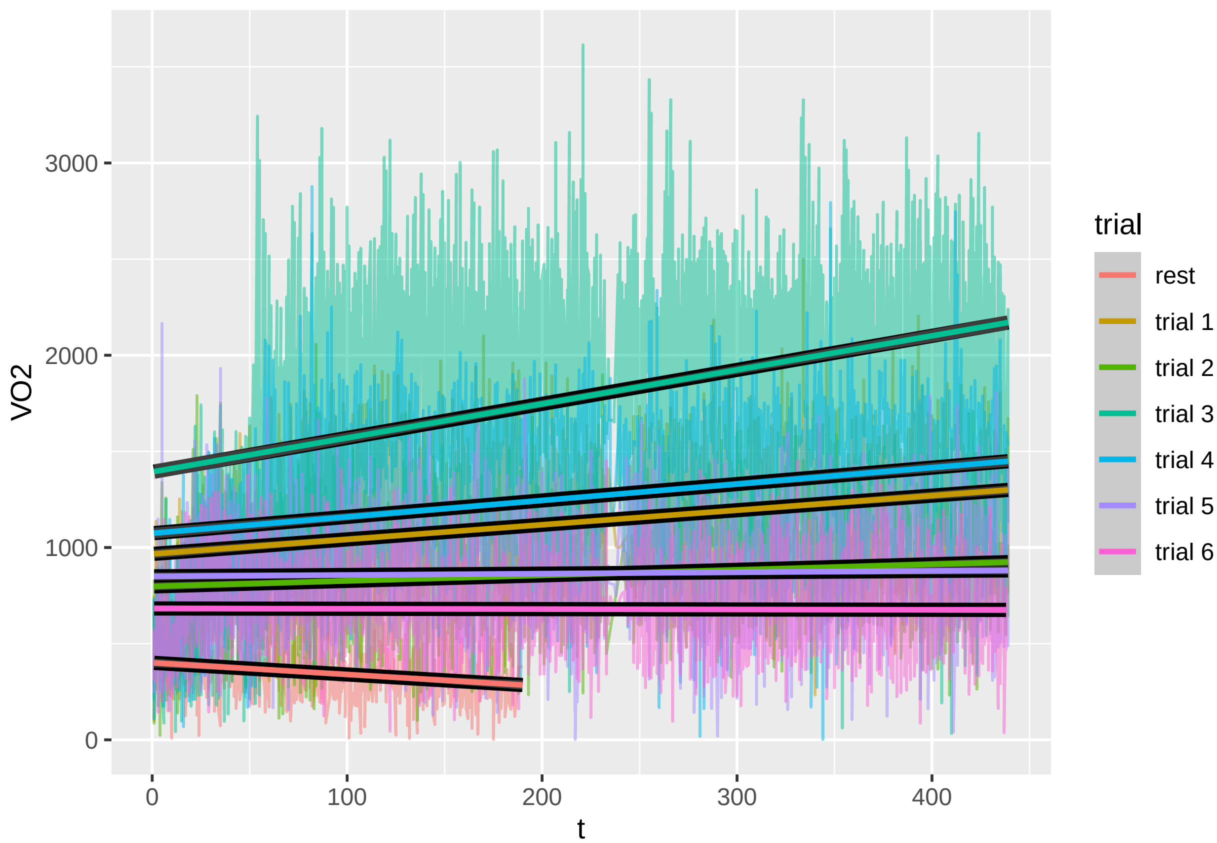
[1] CI = Confidence Interval

1.3 Mixed Models

```
data.all=data.all[data.all$t<440,]

ggplot(data.all,aes(x=t,y=VO2,color=trial,group=trial))+
  geom_line(alpha=.5)+
  geom_smooth(method = 'lm',color="black",se=F,size=2.5)+
  geom_smooth(method = 'lm')
```

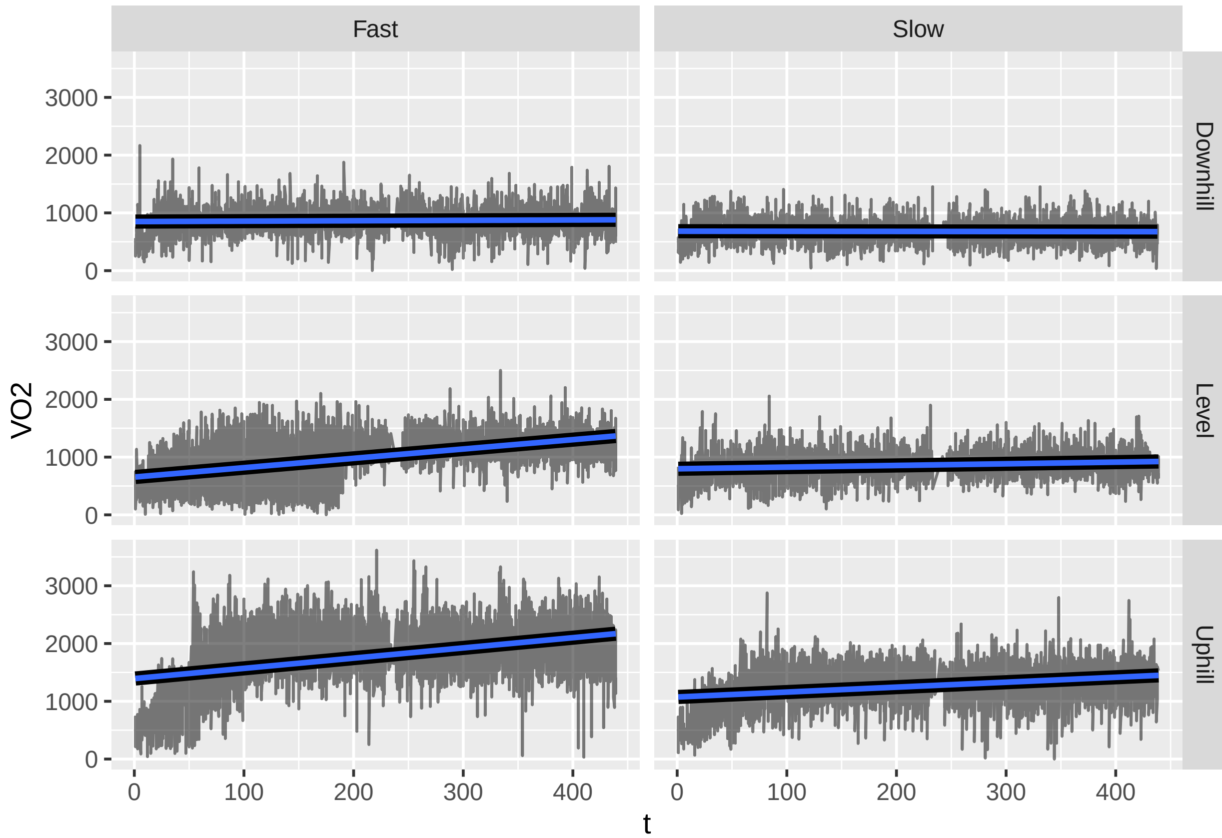
```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```




```
data.all$Level="Level"
data.all$Level[data.all$trial=="trial 3" | data.all$trial=="trial 4"]="Uphill"
data.all$Level[data.all$trial=="trial 5" | data.all$trial=="trial 6"]="Downhill"

data.all$Speed="Fast"
data.all$Speed[data.all$trial=="trial 2" | data.all$trial=="trial 4" | data.all$trial=="trial 6"]="Slow

ggplot(data.all,aes(x=t,y=VO2))+
  geom_line(alpha=.5)+
  geom_smooth(method = 'lm',color="black",se=F,size=2.5)+
  geom_smooth(method = 'lm')+
  facet_grid(Level~Speed)
```

```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```




```
ggplot(data.all,aes(x=t,y=VO2,color=Speed))+
  geom_line(alpha=.5)+
  geom_smooth(method = 'lm')+
  facet_wrap(~Level)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```


```r
fit.lme=lme(VO2~t*Speed+t*Level+Age+Gender,
            random = ~1|Sub,
            method = "ML",
            data.all)

Anova(fit.lme)
```

Chisq	Df	Pr(>Chisq)
1.84e+03	1	0
3.15e+03	1	0
1.63e+04	2	0
1.9	1	0.168
1.1	1	0.295
436	1	9.55e-97
761	2	6.16e-166

```r
vif(fit.lme)
```

```
##                GVIF Df GVIF^(1/(2*Df))
```

```
## t         4.266512  1        2.065554
## Speed     3.923103  1        1.980682
## Level    15.218707  2        1.975124
## Age       1.304089  1        1.141967
## Gender    1.304072  1        1.141960
## t:Speed   4.853385  1        2.203040
## t:Level  23.596949  2        2.204012
```

```r
summary(fit.lme)
```

```
## Linear mixed-effects model fit by maximum likelihood
##   Data: data.all
##        AIC      BIC    logLik
##   209435.1 209526.2 -104705.6
##
## Random effects:
##  Formula: ~1 | Sub
##         (Intercept) Residual
## StdDev:    154.0267 305.8998
##
## Fixed effects:  VO2 ~ t * Speed + t * Level + Age + Gender
##                    Value Std.Error    DF   t-value p-value
## (Intercept)     433.3566 231.10377 14635   1.87516  0.0608
## t                 0.4348   0.04093 14635  10.62256  0.0000
## SpeedSlow      -105.0763  10.05381 14635 -10.45140  0.0000
## LevelLevel      -70.9294  12.24631 14635  -5.79190  0.0000
## LevelUphill     458.4397  12.63535 14635  36.28232  0.0000
## Age              14.1198  10.25215    10   1.37725  0.1985
## GenderM         102.6570  98.06637    10   1.04681  0.3198
## t:SpeedSlow      -0.8324   0.03990 14635 -20.86523  0.0000
## t:LevelLevel      1.0083   0.04919 14635  20.49767  0.0000
## t:LevelUphill     1.3102   0.04942 14635  26.50968  0.0000
##  Correlation:
##               (Intr) t      SpdSlw LvlLvl LvlUph Age    GendrM t:SpdS t:LvlL
## t             -0.039
## SpeedSlow     -0.021  0.419
## LevelLevel    -0.031  0.599  0.099
## LevelUphill   -0.027  0.551  0.013  0.538
## Age           -0.961  0.000  0.000 -0.001 -0.001
## GenderM        0.289  0.000  0.000 -0.001 -0.001 -0.483
## t:SpeedSlow    0.019 -0.480 -0.862 -0.073 -0.010 -0.001  0.000
## t:LevelLevel   0.026 -0.666 -0.073 -0.858 -0.456  0.000  0.000  0.054
## t:LevelUphill  0.025 -0.643 -0.011 -0.468 -0.867  0.000  0.000  0.014  0.531
##
## Standardized Within-Group Residuals:
##         Min          Q1         Med          Q3         Max
## -5.92814345 -0.51964736  0.05480858  0.56898220  5.64026503
##
## Number of Observations: 14655
## Number of Groups: 13
```

```r
library(emmeans)
```

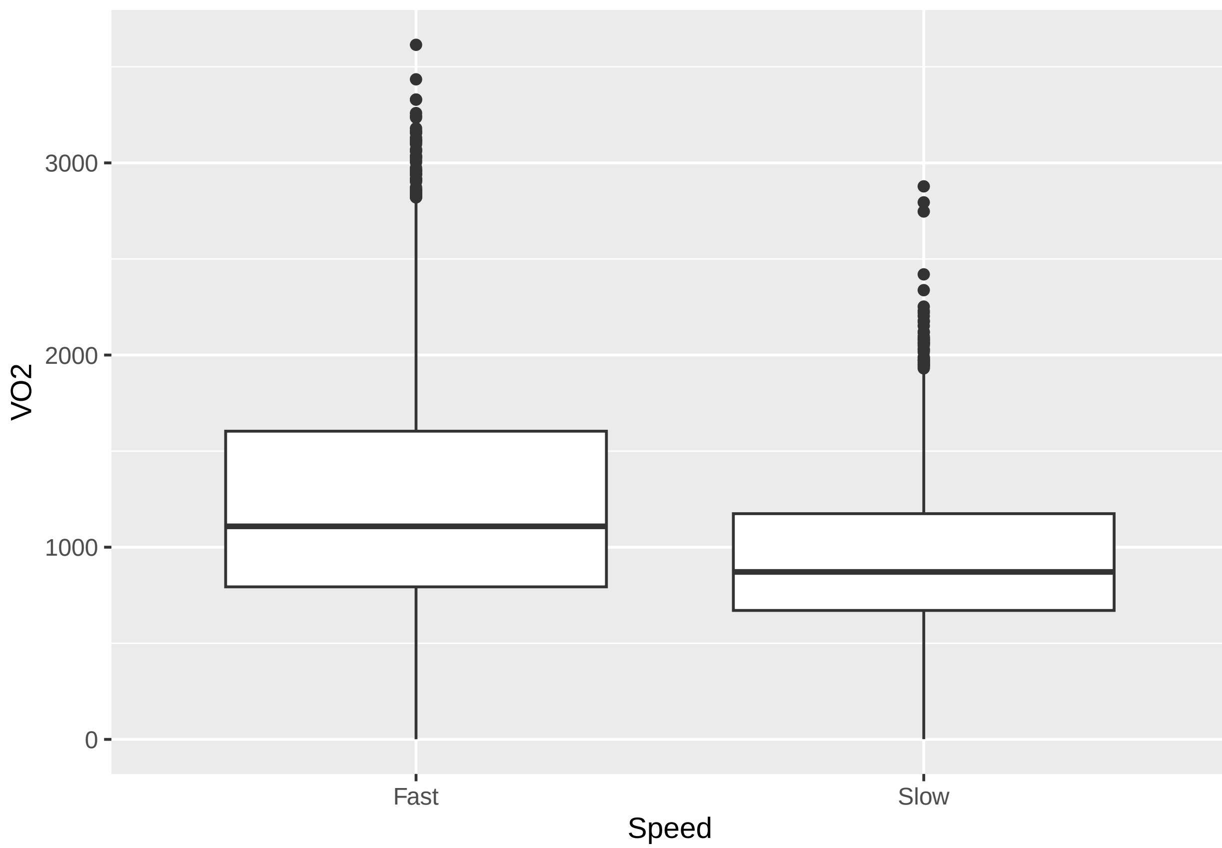
```
## Welcome to emmeans.
## Caution: You lose important information if you filter this package's results.
## See '? untidy'
```

```r
#Calculate marginal (adjusted) means for level and speed
#This gives us the average adjusted VO2 by speed and level
emmeans(fit.lme,pairwise ~ Speed)
```

```
## NOTE: Results may be misleading due to involvement in interactions
```

```
## $emmeans
##  Speed emmean SE df lower.CL upper.CL
##  Fast    1213 43 10     1117     1309
##  Slow     928 43 10      832     1024
##
## Results are averaged over the levels of: Level, Gender
## Degrees-of-freedom method: containment
## Confidence level used: 0.95
##
## $contrasts
##  contrast    estimate  SE    df t.ratio p.value
##  Fast - Slow      285 5.1 14635  55.911  <.0001
##
## Results are averaged over the levels of: Level, Gender
## Degrees-of-freedom method: containment
```

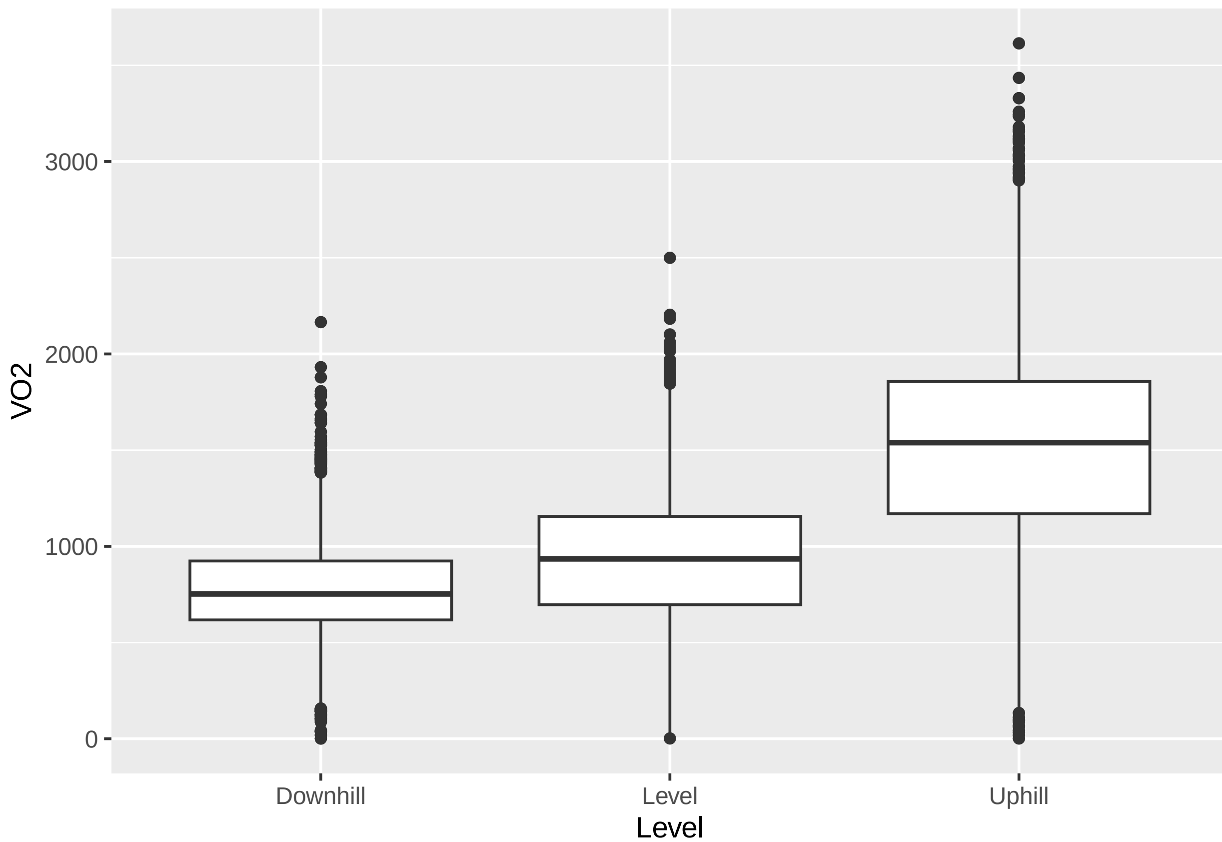
```r
ggplot(data.all,aes(x=Speed,y=VO2))+
  geom_boxplot()
```




```
emmeans(fit.lme,pairwise ~ Level)
```

```
## NOTE: Results may be misleading due to involvement in interactions

## $emmeans
##  Level    emmean   SE df lower.CL upper.CL
##  Downhill    775 43.1 10      679      871
##  Level       922 43.1 10      826     1018
##  Uphill     1516 43.1 10     1420     1612
##
## Results are averaged over the levels of: Speed, Gender
## Degrees-of-freedom method: containment
## Confidence level used: 0.95
##
## $contrasts
##  contrast          estimate   SE    df  t.ratio p.value
##  Downhill - Level      -147 6.30 14635  -23.325  <.0001
##  Downhill - Uphill     -742 6.31 14635 -117.557  <.0001
##  Level - Uphill        -595 6.11 14635  -97.355  <.0001
##
## Results are averaged over the levels of: Speed, Gender
## Degrees-of-freedom method: containment
## P value adjustment: tukey method for comparing a family of 3 estimates
```

```r
ggplot(data.all,aes(x=Level,y=VO2))+
  geom_boxplot()
```



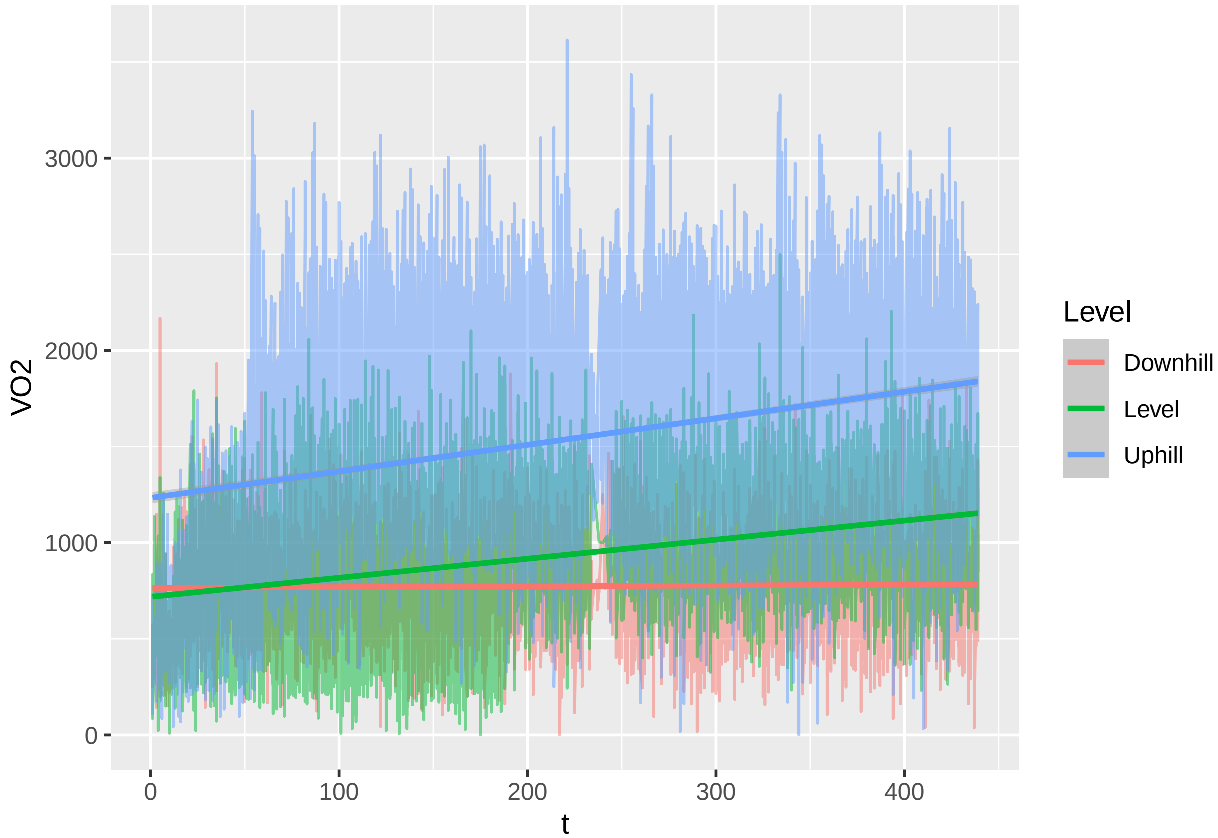

```r
#emtrends will return the pairwise comparisons between the slopes of each
#condition across time (t)
emtrends(fit.lme,pairwise ~ Level,var = "t")
```

```
## $emtrends
##  Level     t.trend     SE    df lower.CL upper.CL
##  Downhill   0.0186 0.0359 14635  -0.0518   0.0889
##  Level      1.0269 0.0336 14635   0.9610   1.0928
##  Uphill     1.3288 0.0340 14635   1.2621   1.3954
##
## Results are averaged over the levels of: Speed, Gender
## Degrees-of-freedom method: containment
## Confidence level used: 0.95
##
## $contrasts
##  contrast           estimate     SE    df t.ratio p.value
##  Downhill - Level     -1.008 0.0492 14635 -20.498  <.0001
##  Downhill - Uphill    -1.310 0.0494 14635 -26.510  <.0001
##  Level - Uphill       -0.302 0.0478 14635  -6.319  <.0001
##
## Results are averaged over the levels of: Speed, Gender
## Degrees-of-freedom method: containment
```

```
## P value adjustment: tukey method for comparing a family of 3 estimates
```

```r
ggplot(data.all,aes(x=t,y=VO2,color=Level))+
  geom_line(alpha=.5)+
  geom_smooth(method = 'lm')
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

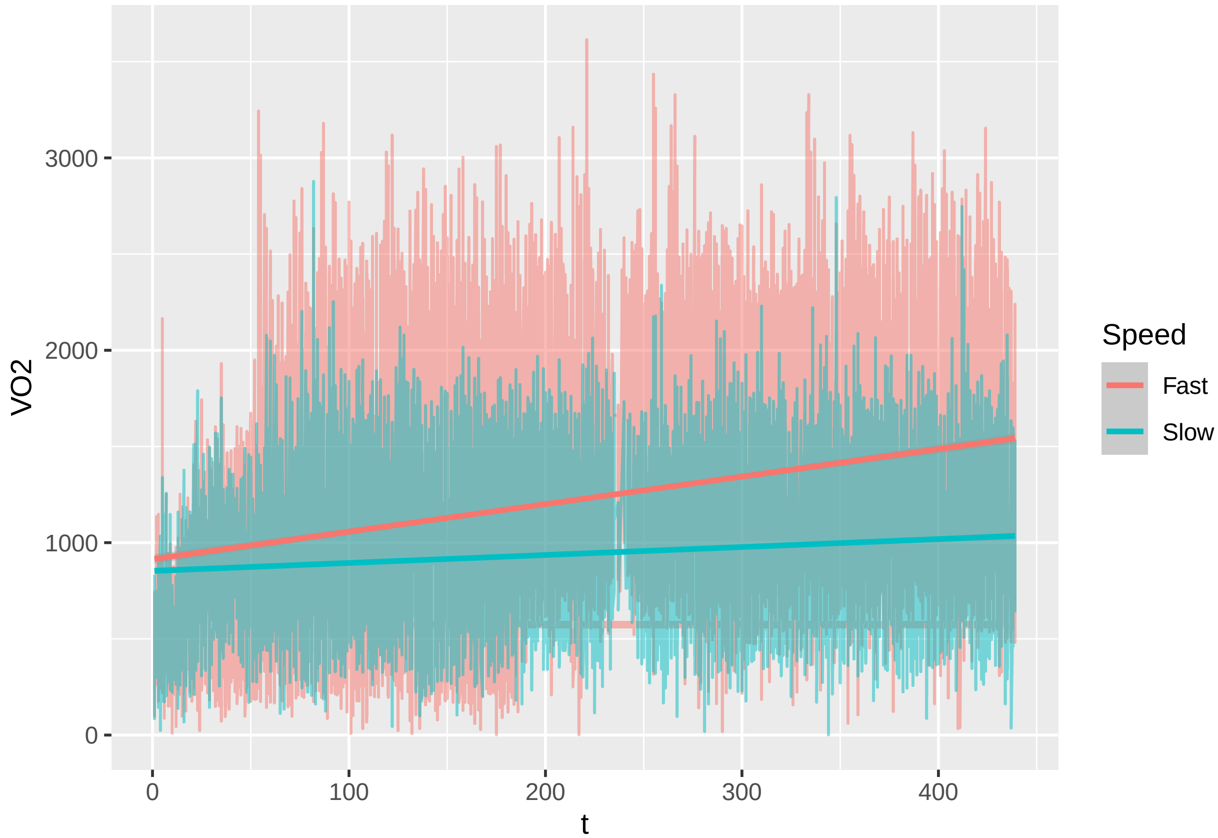



```r
emtrends(fit.lme,pairwise ~ Speed,var = "t")
```

```
## $emtrends
##  Speed t.trend     SE    df lower.CL upper.CL
##  Fast    1.208 0.0272 14635    1.154    1.261
##  Slow    0.375 0.0292 14635    0.318    0.432
##
## Results are averaged over the levels of: Level, Gender
## Degrees-of-freedom method: containment
## Confidence level used: 0.95
##
## $contrasts
##  contrast    estimate     SE    df t.ratio p.value
##  Fast - Slow    0.832 0.0399 14635  20.865  <.0001
##
## Results are averaged over the levels of: Level, Gender
## Degrees-of-freedom method: containment
```

```
ggplot(data.all,aes(x=t,y=VO2,color=Speed))+
  geom_line(alpha=.5)+
  geom_smooth(method = 'lm')
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



1.4 Model Comparison

Our mixed effects model had the relationship between t and VO2 as linear.

However, that may not be the model of best fit. A non-linear model may fit better.

#But how would we determine this to be the case?

```
#We can either compare model metrics or run ANOVA between nested models
AIC(fit.lme)
```

```
## [1] 209435.1
```

```
fit.lme.log=lme(VO2~log(t)*Speed+
                  log(t)*Level+Age+Gender,
            random = ~1|Sub,
            method = "ML",
            data.all)

Anova(fit.lme.log)
```

Chisq	Df	Pr(>Chisq)
3.3e+03	1	0
3.48e+03	1	0
1.82e+04	2	0
1.86	1	0.173
1.08	1	0.299
568	1	1.91e-125
1.37e+03	2	1.49e-298

```
vif(fit.lme.log)
```

```
##                      GVIF Df GVIF^(1/(2*Df))
## log(t)           4.416124  1        2.101458
## Speed           28.181549  1        5.308630
## Level          790.594611  2        5.302595
## Age              1.304077  1        1.141962
## Gender           1.304062  1        1.141955
## log(t):Speed    29.193456  1        5.403097
## log(t):Level   846.776771  2        5.394389
```

```
summary(fit.lme.log)
```

```
## Linear mixed-effects model fit by maximum likelihood
##   Data: data.all
##        AIC      BIC    logLik
##   207725.4 207816.5 -103850.7
##
## Random effects:
##  Formula: ~1 | Sub
##         (Intercept) Residual
## StdDev:    153.7641 288.5519
##
## Fixed effects:  VO2 ~ log(t) * Speed + log(t) * Level + Age + Gender
##                         Value Std.Error    DF   t-value p-value
## (Intercept)          134.9906 231.95374 14635   0.58197  0.5606
## log(t)                77.9411   5.11454 14635  15.23911  0.0000
## SpeedSlow            311.0288  25.41811 14635  12.23651  0.0000
## LevelLevel          -423.9430  30.83541 14635 -13.74858  0.0000
## LevelUphill         -423.3711  32.18869 14635 -13.15279  0.0000
## Age                   13.9486  10.23251    10   1.36317  0.2027
## GenderM              101.7207  97.88005    10   1.03924  0.3232
## log(t):SpeedSlow    -117.0505   4.91491 14635 -23.81540  0.0000
## log(t):LevelLevel    112.5951   5.98895 14635  18.80049  0.0000
## log(t):LevelUphill   229.1480   6.19299 14635  37.00121  0.0000
##  Correlation:
```

```
##                      (Intr) log(t) SpdSlw LvlLvl LvlUph Age    GendrM l():SS
## log(t)               -0.113
## SpeedSlow            -0.056  0.484
## LevelLevel           -0.080  0.689  0.108
## LevelUphill          -0.072  0.620  0.013  0.538
## Age                  -0.956  0.000  0.000  0.000  0.000
## GenderM               0.287  0.000  0.000  0.000  0.000 -0.483
## log(t):SpeedSlow      0.055 -0.491 -0.982 -0.099 -0.013  0.000  0.000
## log(t):LevelLevel     0.078 -0.693 -0.099 -0.981 -0.525  0.000  0.000  0.092
## log(t):LevelUphill    0.071 -0.633 -0.013 -0.530 -0.983  0.000  0.000  0.014
##                      l():LL
## log(t)
## SpeedSlow
## LevelLevel
## LevelUphill
## Age
## GenderM
## log(t):SpeedSlow
## log(t):LevelLevel
## log(t):LevelUphill  0.536
##
## Standardized Within-Group Residuals:
##         Min          Q1         Med          Q3         Max
## -6.11054106 -0.49904101  0.05688234  0.56342255  5.74795809
##
## Number of Observations: 14655
## Number of Groups: 13
```

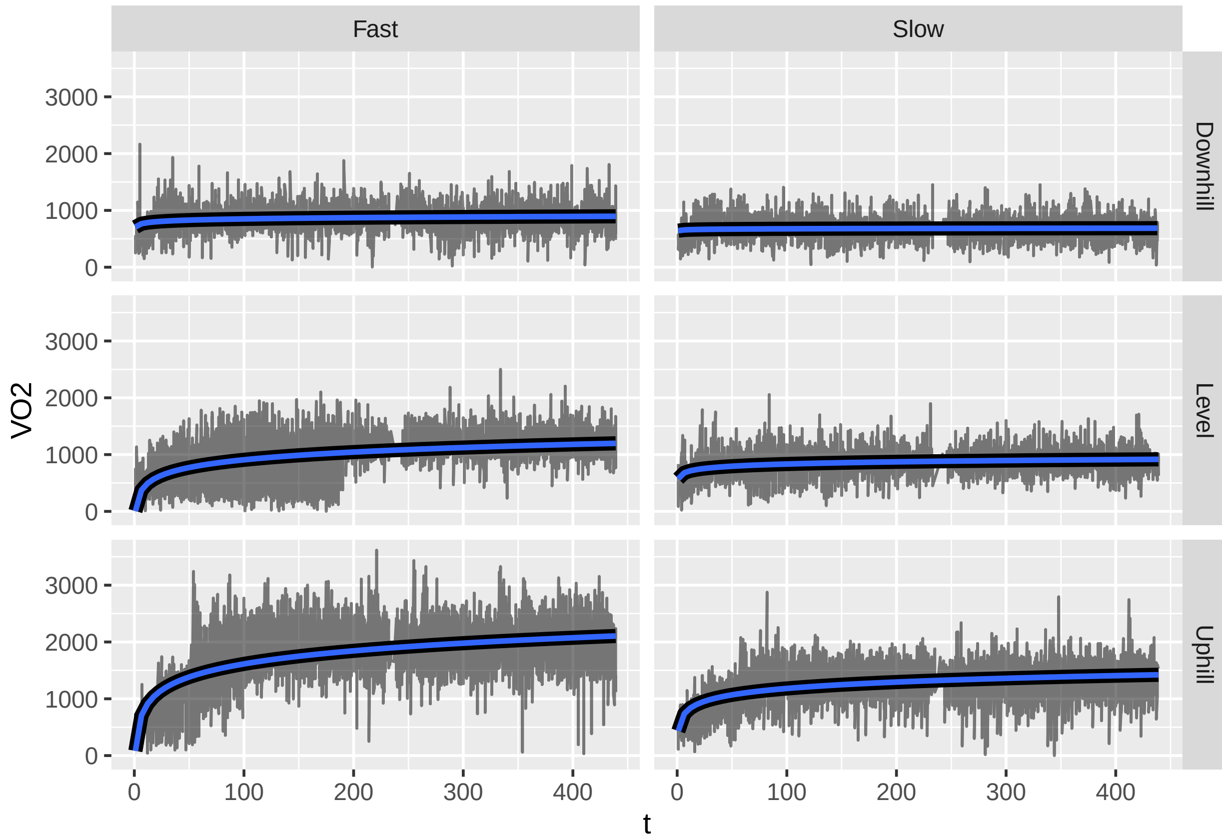
```r
AIC(fit.lme.log)
```

```
## [1] 207725.4
```

```r
anova(fit.lme,fit.lme.log)
```

	Model	df	AIC
~ t * Speed + t * Level + Age + Gender, data = data.all, random = ~1 | Sub, method = "ML")	1	12	2.09e+05
~ log(t) * Speed + log(t) * Level + Age + Gender, data = data.all, random = ~1 | Sub, method = "ML")	2	12	2.08e+05

```r
ggplot(data.all,aes(x=t,y=VO2))+
  geom_line(alpha=.5)+
  geom_smooth(method = 'lm',formula = "y~log(x)",color="black",se=F,size=2.5)+
  geom_smooth(method = 'lm',formula = "y~log(x)")+
  facet_grid(Level~Speed)
```




```r
#What if instead of a log we model it as a 2nd degree polynomial
fit.lme.poly2=lme(VO2~poly(t,2,raw=TRUE)*Speed+
                    poly(t,2,raw=TRUE)*Level+Age+Gender,
             random = ~1|Sub,
             method = "ML",
             data.all)

Anova(fit.lme.poly2)
```

Chisq	Df	Pr(>Chisq)
3.11e+03	2	0
3.55e+03	1	0
1.82e+04	2	0
1.85	1	0.174
1.07	1	0.3
644	2	1.67e-140
1.48e+03	4	2.2e-319

```r
vif(fit.lme.poly2)
```

```
##                                      GVIF Df GVIF^(1/(2*Df))
## poly(t, 2, raw = TRUE)        18.307246  2        2.068501
## Speed                          9.141569  1        3.023503
## Level                         82.588944  2        3.014605
## Age                            1.304078  1        1.141962
## Gender                         1.304062  1        1.141955
## poly(t, 2, raw = TRUE):Speed  18.953302  2        2.086514
## poly(t, 2, raw = TRUE):Level 333.075849  4        2.066893
```

```r
summary(fit.lme.poly2)
```

```
## Linear mixed-effects model fit by maximum likelihood
##   Data: data.all
##        AIC      BIC    logLik
##   207730.9 207852.4 -103849.5
##
## Random effects:
##  Formula: ~1 | Sub
##         (Intercept) Residual
## StdDev:    153.9359 288.5273
##
## Fixed effects:  VO2 ~ poly(t, 2, raw = TRUE) * Speed + poly(t, 2, raw = TRUE) *      Level + Age + Ge
##                                      Value Std.Error    DF    t-value
## (Intercept)                       338.1266 231.20204 14631   1.462472
## poly(t, 2, raw = TRUE)1             1.7887   0.15917 14631  11.238002
## poly(t, 2, raw = TRUE)2            -0.0031   0.00035 14631  -8.734028
## SpeedSlow                          28.3196  14.47749 14631   1.956112
## LevelLevel                       -142.0628  17.55675 14631  -8.091634
## LevelUphill                       123.2148  18.24374 14631   6.753814
## Age                                13.9320  10.24530    10   1.359839
## GenderM                           101.4784  98.00240    10   1.035468
## poly(t, 2, raw = TRUE)1:SpeedSlow  -2.6442   0.15411 14631 -17.158117
## poly(t, 2, raw = TRUE)2:SpeedSlow   0.0041   0.00034 14631  12.072959
## poly(t, 2, raw = TRUE)1:LevelLevel  2.0999   0.18915 14631  11.101960
## poly(t, 2, raw = TRUE)2:LevelLevel -0.0026   0.00042 14631  -6.176224
## poly(t, 2, raw = TRUE)1:LevelUphill 5.7534   0.19203 14631  29.960630
## poly(t, 2, raw = TRUE)2:LevelUphill -0.0100  0.00042 14631 -23.686760
##                                   p-value
## (Intercept)                        0.1436
## poly(t, 2, raw = TRUE)1            0.0000
## poly(t, 2, raw = TRUE)2            0.0000
## SpeedSlow                          0.0505
## LevelLevel                         0.0000
## LevelUphill                        0.0000
## Age                                0.2037
## GenderM                            0.3248
## poly(t, 2, raw = TRUE)1:SpeedSlow  0.0000
## poly(t, 2, raw = TRUE)2:SpeedSlow  0.0000
## poly(t, 2, raw = TRUE)1:LevelLevel 0.0000
## poly(t, 2, raw = TRUE)2:LevelLevel 0.0000
```

```
## poly(t, 2, raw = TRUE)1:LevelUphill  0.0000
## poly(t, 2, raw = TRUE)2:LevelUphill  0.0000
##  Correlation:
##                                     (Intr) pl(,2,r=TRUE)1 pl(,2,r=TRUE)2 SpdSlw
## poly(t, 2, raw = TRUE)1             -0.056
## poly(t, 2, raw = TRUE)2              0.049 -0.970
## SpeedSlow                           -0.031  0.427         -0.371
## LevelLevel                          -0.045  0.607         -0.525          0.109
## LevelUphill                         -0.040  0.551         -0.479          0.012
## Age                                 -0.960  0.000          0.000         -0.001
## GenderM                              0.289  0.000          0.000          0.000
## poly(t, 2, raw = TRUE)1:SpeedSlow    0.027 -0.487          0.471         -0.870
## poly(t, 2, raw = TRUE)2:SpeedSlow   -0.023  0.469         -0.482          0.755
## poly(t, 2, raw = TRUE)1:LevelLevel   0.039 -0.678          0.656         -0.081
## poly(t, 2, raw = TRUE)2:LevelLevel  -0.033  0.650         -0.668          0.065
## poly(t, 2, raw = TRUE)1:LevelUphill  0.036 -0.640          0.623         -0.012
## poly(t, 2, raw = TRUE)2:LevelUphill -0.031  0.624         -0.645          0.011
##                                     LvlLvl LvlUph Age    GendrM p(,2,r=TRUE)1:S
## poly(t, 2, raw = TRUE)1
## poly(t, 2, raw = TRUE)2
## SpeedSlow
## LevelLevel
## LevelUphill                          0.536
## Age                                  0.000 -0.001
## GenderM                              0.000  0.000 -0.483
## poly(t, 2, raw = TRUE)1:SpeedSlow   -0.082 -0.012  0.000  0.000
## poly(t, 2, raw = TRUE)2:SpeedSlow    0.066  0.011 -0.001  0.000 -0.970
## poly(t, 2, raw = TRUE)1:LevelLevel  -0.868 -0.460  0.000  0.000  0.073
## poly(t, 2, raw = TRUE)2:LevelLevel   0.752  0.396  0.000  0.000 -0.063
## poly(t, 2, raw = TRUE)1:LevelUphill -0.470 -0.872  0.000  0.000  0.016
## poly(t, 2, raw = TRUE)2:LevelUphill  0.410  0.757  0.000  0.000 -0.016
##                                     p(,2,r=TRUE)2:S p(,2,r=TRUE)1:LL
## poly(t, 2, raw = TRUE)1
## poly(t, 2, raw = TRUE)2
## SpeedSlow
## LevelLevel
## LevelUphill
## Age
## GenderM
## poly(t, 2, raw = TRUE)1:SpeedSlow
## poly(t, 2, raw = TRUE)2:SpeedSlow
## poly(t, 2, raw = TRUE)1:LevelLevel  -0.063
## poly(t, 2, raw = TRUE)2:LevelLevel   0.056          -0.969
## poly(t, 2, raw = TRUE)1:LevelUphill -0.016           0.533
## poly(t, 2, raw = TRUE)2:LevelUphill  0.017          -0.520
##                                     p(,2,r=TRUE)2:LL p(,2,r=TRUE)1:LU
## poly(t, 2, raw = TRUE)1
## poly(t, 2, raw = TRUE)2
## SpeedSlow
## LevelLevel
## LevelUphill
## Age
## GenderM
## poly(t, 2, raw = TRUE)1:SpeedSlow
```

```
## poly(t, 2, raw = TRUE)2:SpeedSlow
## poly(t, 2, raw = TRUE)1:LevelLevel
## poly(t, 2, raw = TRUE)2:LevelLevel
## poly(t, 2, raw = TRUE)1:LevelUphill -0.514
## poly(t, 2, raw = TRUE)2:LevelUphill  0.533           -0.970
##
## Standardized Within-Group Residuals:
##         Min          Q1         Med          Q3         Max
## -6.06653766 -0.51167210  0.03269341  0.56566534  6.11597394
##
## Number of Observations: 14655
## Number of Groups: 13
```

```
AIC(fit.lme.poly2)
```

```
## [1] 207730.9
```

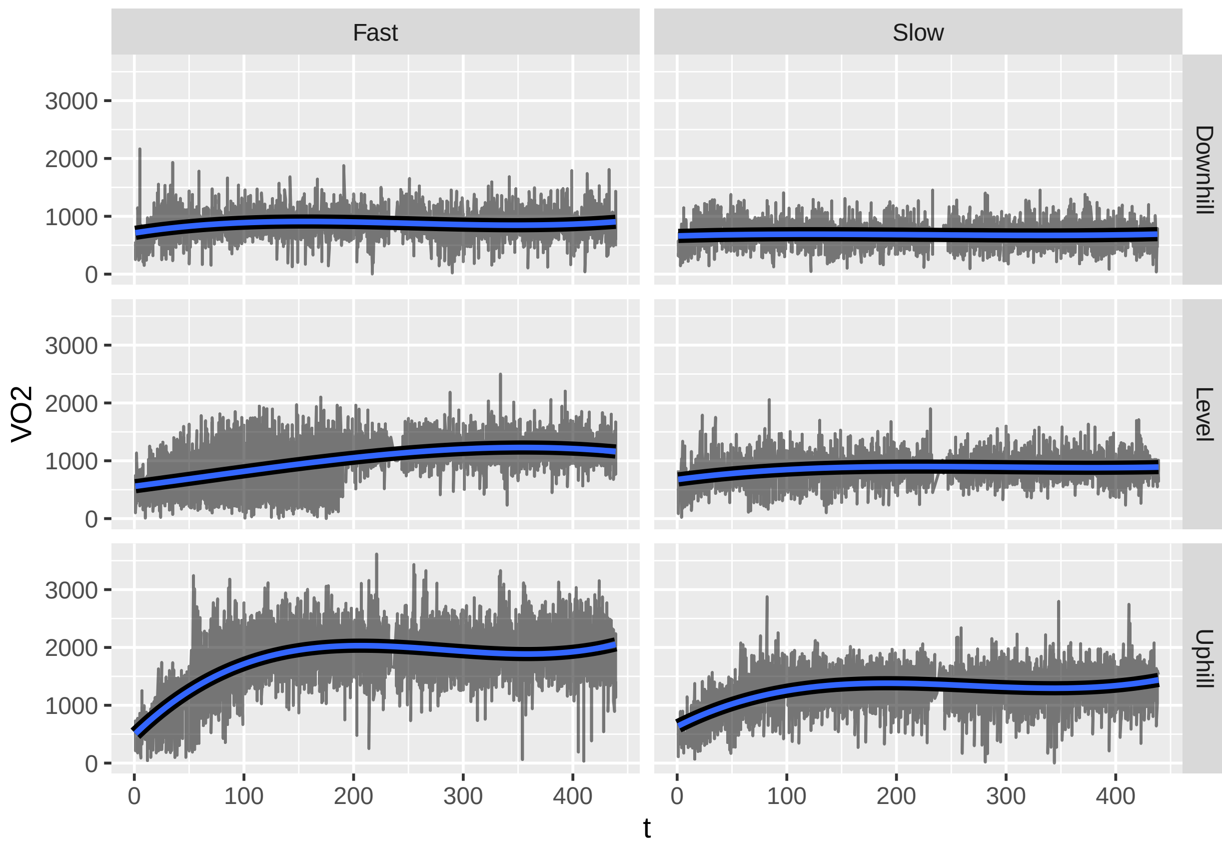
```
emtrends(fit.lme.poly2,pairwise ~ Level,var="t")
```

```
## $emtrends
##  Level    t.trend     SE    df lower.CL upper.CL
##  Downhill  0.0322 0.0340 14631  -0.0344   0.0987
##  Level     1.0128 0.0318 14631   0.9505   1.0750
##  Uphill    1.4726 0.0323 14631   1.4094   1.5359
##
## Results are averaged over the levels of: Speed, Gender
## Degrees-of-freedom method: containment
## Confidence level used: 0.95
##
## $contrasts
##  contrast          estimate     SE    df t.ratio p.value
##  Downhill - Level    -0.981 0.0465 14631 -21.085  <.0001
##  Downhill - Uphill   -1.440 0.0468 14631 -30.765  <.0001
##  Level - Uphill      -0.460 0.0453 14631 -10.163  <.0001
##
## Results are averaged over the levels of: Speed, Gender
## Degrees-of-freedom method: containment
## P value adjustment: tukey method for comparing a family of 3 estimates
```

```
#In this case a log model is just as good as a poly model and a little simpler
anova(fit.lme,fit.lme.poly2,fit.lme.log)
```

	Model	df	AIC
ender, data = data.all, random = ~1 | Sub, method = "ML")	1	12	2.09e+05
+ poly(t, 2, raw = TRUE) * Level + Age + Gender, data = data.all, random = ~1 | Sub, method = "ML")	2	16	2.08e+05
Age + Gender, data = data.all, random = ~1 | Sub, method = "ML")	3	12	2.08e+05

```r
#However, there may exist a model that is more complex and a better fit.
ggplot(data.all,aes(x=t,y=VO2))+
  geom_line(alpha=.5)+
  geom_smooth(method = 'lm',formula = "y~poly(x,3)",color="black",se=F,size=2.5)+
  geom_smooth(method = 'lm',formula = "y~poly(x,3)")+
  facet_grid(Level~Speed)
```




```r
#At this point we would move to a non-linear mixed effects model.
```