R markdown of BeHo initial analyses

2023-10-16

Project description

Much evidence suggests that the gut microbiome influences animal behaviour. Nevertheless, many behavioural traits have not been investigated yet, in particular those unassociated with diseases. Thus, in this study, we infer the behaviour dominance in J:DO laboratory mice (mus musculus) to test if the gut microbiome plays a role herein during different experimental treatments. Over the course of 14 weeks, we measured dominance of 80 mice in total, 40 of each sex, using a tube test to rank the dominance within a cage. Five males or five females were housed together per cage, and a hierarchy of dominance was therefore created ranking the individuals from 1 to 5, where 1 signified "most dominant" and 5 as "most submissive". We further wanted to explore if the gut microbiome affects dominance under different biological conditions, such as changes in temperature or diet. Hence, the mice were exposed to five specific treatments, all known to interfere with the gut microbiome, each following a four day resting period for the mice to restore the potential unbalance caused (e.g. physiological or gut microbial). The order of treatments was i) an exposure to heat (temperature set to 34 degrees celsius), ii) an exposure to cold (temperature set to 14 degrees celsius), iii) a dietary change, where tryptophan was removed, iv) an antibiotic and antifungal treatment, and lastly, v) a fecal microbiota transplantation, where, among other, the feces from the most dominant animal was given to the most submissive, and vice versa, within each cage. However, one cage per sex were not exposed to any changes, and thus, they serve as control cages to examine what the gut microbes may have had of an effect on dominance without any disturbances.

Read more about the BeHo mice project: https://docs.google.com/document/d/1Ra6K9tJBcnQmh23X5Pv3lZoZJmxm84B3S
diuWY/edit?usp=sharing

Libraries are loaded, the environment set and files read in

The libraries loaded. Not all of them are used though. library(tidyverse) library(reshape2) library(tidyr) library(effects) library(ggplot2) library(sjPlot) library(vioplot) library(ggpubr) library(PerformanceAnalytics)

Mice who died during the experiment before it ended have zeros (0) written as tube test results, which are misguiding. Thus, the zeroes are replaced with NA.

To explain the columns of the dataframes that might not be intuitive:

R1, R2, R3, R4 and R5 = the replicates of tube test per treatment. Each replicate represents a day. Hence, if a treatment has three replicates, then they were performed over three days. All cages were tested in one day and the repeated the next. The mice were tested in the last days of the experiment (except for FMT). For instance, in the diet treatment the last three days = three tube test replicates. One per day. Thus, the diet treatment lasted 8 days and the tube tests were conducted on day 6, 7 and 8. However, the FMT treatment there are five replicates in total, because we wanted to test if the dominance changes during the FMT treatment. Thus, we did tube tests FMT day 3, 6, 9, 10 and 11. Thus, FMT day 9-11 are the three consecutive replicates like the previous treatments.

Wins_per_rep = the number of wins an individual mouse has within a replicate. The number of wins can vary from 0 to 4. As all five mice within a cage "fight" each other. However, the number of wins don't necessarily dsitribute as 0, 1, 2, 3 and 4 for each mouse within a cage.

OPT = optimal conditions. The mice were household in 24 degrees celsius and no disturbances were made. Alias for this "treatment": Acclimation. Duration: 4 weeks. HEAT = The experimental mice were exposed to 34 degrees celsius. Duration: 12 days. COLD = The experimental mice were exposed to 14 degrees celsius. Duration: 12 days. DIET = The experimental mice were exposed to a tryptophan depleted diet. Duration: 8 days. ANTI = The experimental mice were exposed to antibiotics and anti-fungals to remove the existing gut microbes. The mice were given antibiotics for the first five days. Duration: 8 days. FMT = The experimental mice were exposed to fecal microbiota transplantations (FMT). The FMTs were given the first two days. Duration: 11 days.

Dataframe transformation

The files have been read in and are now transformed. I create two dataframes, where tube_test_80 include all 80 mice and tube_test_40 include 40 mice. These 40 mice are the ones that we have genomics and metagenomics data on. The 40 mice are composed of 20 females and 20 males: The male cages are 3, 4, 5 and 6. The female cages are 10, 11, 12 and 13. Cage 4 (male) and 13 (female) are control cages.

```
#Creating new factors to start statistics.
#A factor instead of num, chr or int account for variation between the "categories"
tube_test_80 <- transform(tube_test_80,Replicates=as.factor(Replicates),
                          Phenotype=as.factor(Phenotype),
                          Crate=as.factor(Crate_ID),
                          Sex=as.factor(Sex),
                          Exp_group=as.factor(Exp_group),
                          Cage_ID=as.factor(Cage_ID))
str(tube_test_80) #check if the command worked
```

```
## 'data.frame':    1520 obs. of  11 variables:
##  $ Wins_per_rep : int  2 3 0 4 1 4 1 2 3 0 ...
##  $ Treatments   : Factor w/ 6 levels "OPT","HEAT","COLD",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Replicates   : Factor w/ 5 levels "R1","R2","R3",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Name         : chr  "C01M3_4215" "C01M1_3942" "C01M2_2472" "C01M5_4176" ...
##  $ Arrive_weight: num  19 19.3 18.4 21.8 24 ...
##  $ Phenotype    : Factor w/ 6 levels "Black","Brown",..: 1 2 1 5 6 1 2 2 2 6 ...
##  $ Crate_ID     : chr  "Crate1" "Crate1" "Crate3" "Crate3" ...
##  $ Sex          : Factor w/ 2 levels "Female","Male": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Exp_group    : Factor w/ 2 levels "Control","Treatment": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Cage_ID      : Factor w/ 16 levels "01M","02M","03M",..: 1 1 1 1 1 2 2 2 2 2 ...
##  $ Crate        : Factor w/ 8 levels "Crate1","Crate2",..: 1 1 3 3 3 3 2 2 3 1 ...
```

```
head(tube_test_80)
```

```
##   Wins_per_rep Treatments Replicates       Name Arrive_weight   Phenotype
## 1            2        OPT         R1 C01M3_4215         19.01       Black
## 2            3        OPT         R1 C01M1_3942         19.28       Brown
## 3            0        OPT         R1 C01M2_2472         18.44       Black
## 4            4        OPT         R1 C01M5_4176         21.80 LIGHT BROWN
## 5            1        OPT         R1 C01M4_4146         24.00       White
## 6            4        OPT         R1 C02M3_2832         20.20       Black
##   Crate_ID  Sex Exp_group Cage_ID  Crate
## 1   Crate1 Male Treatment     01M Crate1
## 2   Crate1 Male Treatment     01M Crate1
## 3   Crate3 Male Treatment     01M Crate3
```

```
## 4   Crate3 Male Treatment     01M Crate3
## 5   Crate3 Male Treatment     01M Crate3
## 6   Crate3 Male Treatment     02M Crate3
```

```
#Remove mice that are not among the 40 I'm working with:
#I have mice from cage 3-6 (males) and 10-13 (females)
cages_to_remove <- c("01M","02M","07M","08M","09F","14F","15F","16F")
condition <- tube_test_80$Cage_ID %in% cages_to_remove
tube_test_40 <- tube_test_80[!condition, ]
# Remove rows with missing values (NA) in any column if needing the df like this later
cleaned_tube_test_40 <- drop_na(tube_test_40)

str(tube_test_40)
```

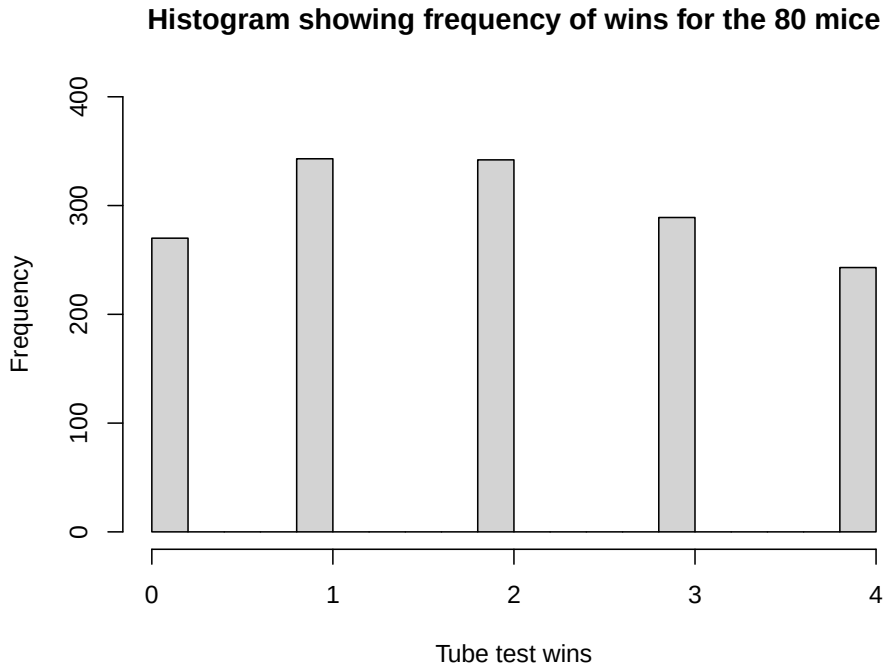
```
## 'data.frame':    760 obs. of  11 variables:
##  $ Wins_per_rep : int  3 0 3 1 3 0 2 4 1 3 ...
##  $ Treatments   : Factor w/ 6 levels "OPT","HEAT","COLD",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Replicates   : Factor w/ 5 levels "R1","R2","R3",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Name         : chr  "C03M2_1576" "C03M1_3856" "C03M3_4011" "C03M4_3083" ...
##  $ Arrive_weight: num  22.2 21 23.9 19 22.6 ...
##  $ Phenotype    : Factor w/ 6 levels "Black","Brown",..: 2 2 6 6 6 6 6 1 2 6 ...
##  $ Crate_ID     : chr  "Crate5" "Crate5" "Crate1" "Crate3" ...
##  $ Sex          : Factor w/ 2 levels "Female","Male": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Exp_group    : Factor w/ 2 levels "Control","Treatment": 2 2 2 2 2 1 1 1 1 1 ...
##  $ Cage_ID      : Factor w/ 16 levels "01M","02M","03M",..: 3 3 3 3 3 4 4 4 4 4 ...
##  $ Crate        : Factor w/ 8 levels "Crate1","Crate2",..: 5 5 1 3 2 2 1 5 2 5 ...
```

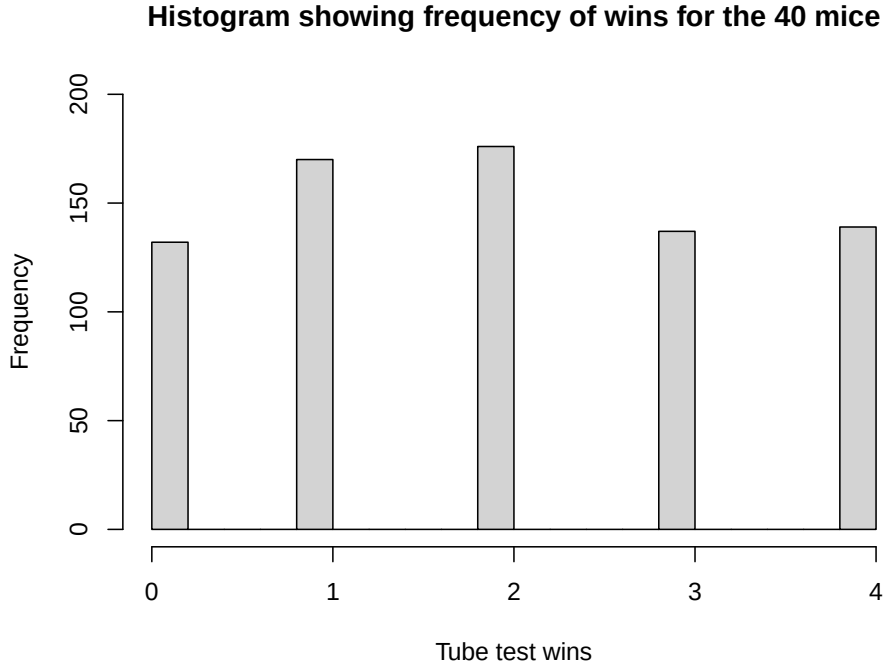
```
head(tube_test_40)
```

```
##    Wins_per_rep Treatments Replicates       Name Arrive_weight Phenotype
## 11            3        OPT         R1 C03M2_1576         22.23     Brown
## 12            0        OPT         R1 C03M1_3856         20.96     Brown
## 13            3        OPT         R1 C03M3_4011         23.90     White
## 14            1        OPT         R1 C03M4_3083         19.02     White
## 15            3        OPT         R1 C03M5_3542         22.55     White
## 16            0        OPT         R1 C04M1_3429         21.66     White
##    Crate_ID  Sex Exp_group Cage_ID  Crate
## 11   Crate5 Male Treatment     03M Crate5
## 12   Crate5 Male Treatment     03M Crate5
## 13   Crate1 Male Treatment     03M Crate1
## 14   Crate3 Male Treatment     03M Crate3
## 15   Crate2 Male Treatment     03M Crate2
## 16   Crate2 Male   Control     04M Crate2
```

Histograms of the dataframes

I plot histograms of both dataframes to visualize the distribution of total number of tube test wins. The data (number of wins) is not normal distributed, but this was expected.





Statistics using lm() and glm()

I have done some statistical tests using linear models and general linear models to test the independent variables against the dependable variable (dominane, number of wins). Many of the tests might not make sense... But perhaps you Inaki have some advice on which ones to use? Because I think glm() works, I just have problems interpreting the results. I hope you can help with this.

First I glimpse at the data:

```r
glimpse(tube_test_40)
```

```
## Rows: 760
## Columns: 11
## $ Wins_per_rep  <int> 3, 0, 3, 1, 3, 0, 2, 4, 1, 3, 3, 1, 2, 4, 0, 1, 2, 4, 0,~
## $ Treatments    <fct> OPT, OPT, OPT, OPT, OPT, OPT, OPT, OPT, OPT, OPT, OPT, O~
## $ Replicates    <fct> R1, R1, R1, R1, R1, R1, R1, R1, R1, R1, R1, R1, R1, R1, ~
## $ Name          <chr> "C03M2_1576", "C03M1_3856", "C03M3_4011", "C03M4_3083", ~
## $ Arrive_weight <dbl> 22.23, 20.96, 23.90, 19.02, 22.55, 21.66, 16.23, 19.66, ~
## $ Phenotype     <fct> Brown, Brown, White, White, White, White, White, Black, ~
## $ Crate_ID      <chr> "Crate5", "Crate5", "Crate1", "Crate3", "Crate2", "Crate~
## $ Sex           <fct> Male, Male, Male, Male, Male, Male, Male, Male, Male, Ma~
## $ Exp_group     <fct> Treatment, Treatment, Treatment, Treatment, Treatment, C~
## $ Cage_ID       <fct> 03M, 03M, 03M, 03M, 03M, 04M, 04M, 04M, 04M, 04M, 05M, 0~
## $ Crate         <fct> Crate5, Crate5, Crate1, Crate3, Crate2, Crate2, Crate1, ~
```

```r
glimpse(tube_test_80)
```

```
## Rows: 1,520
## Columns: 11
## $ Wins_per_rep  <int> 2, 3, 0, 4, 1, 4, 1, 2, 3, 0, 3, 0, 3, 1, 3, 0, 2, 4, 1,~
## $ Treatments    <fct> OPT, OPT, OPT, OPT, OPT, OPT, OPT, OPT, OPT, OPT, OPT, O~
## $ Replicates    <fct> R1, R1, R1, R1, R1, R1, R1, R1, R1, R1, R1, R1, R1, R1, ~
## $ Name          <chr> "C01M3_4215", "C01M1_3942", "C01M2_2472", "C01M5_4176", ~
## $ Arrive_weight <dbl> 19.01, 19.28, 18.44, 21.80, 24.00, 20.20, 17.54, 20.78, ~
## $ Phenotype     <fct> Black, Brown, Black, LIGHT BROWN, White, Black, Brown, B~
## $ Crate_ID      <chr> "Crate1", "Crate1", "Crate3", "Crate3", "Crate3", "Crate~
## $ Sex           <fct> Male, Male, Male, Male, Male, Male, Male, Male, Male, Ma~
## $ Exp_group     <fct> Treatment, Treatment, Treatment, Treatment, Treatment, T~
## $ Cage_ID       <fct> 01M, 01M, 01M, 01M, 01M, 02M, 02M, 02M, 02M, 02M, 03M, 0~
## $ Crate         <fct> Crate1, Crate1, Crate3, Crate3, Crate3, Crate3, Crate2, ~
```

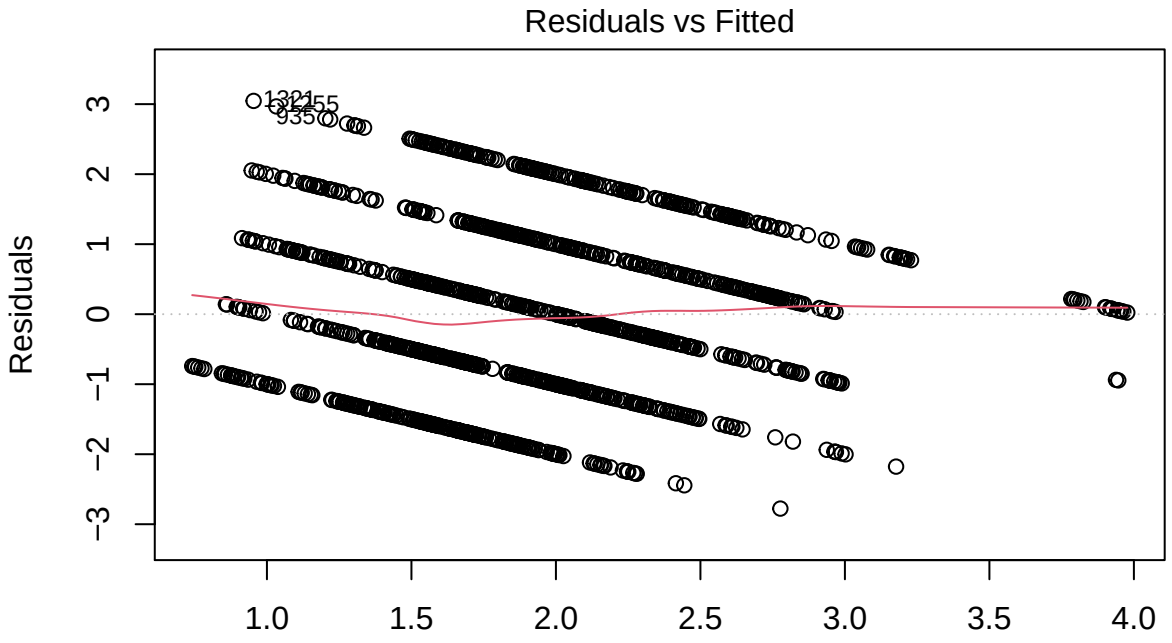
Here I test the data using lm()

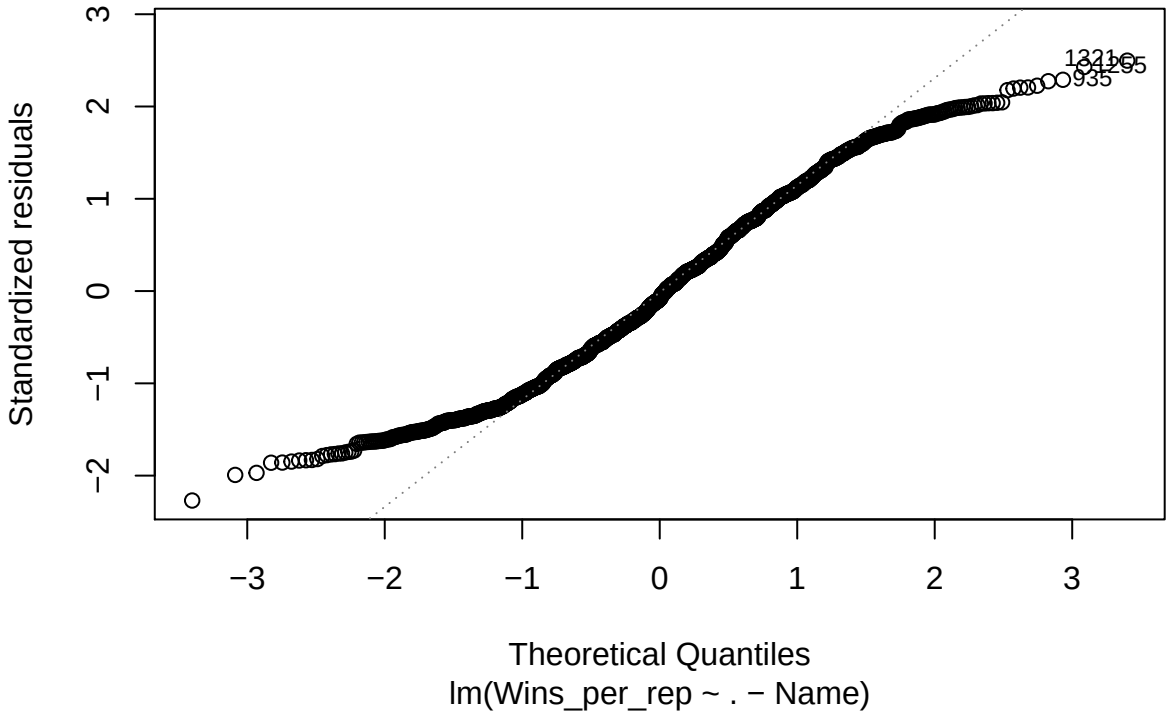
```r
#Because the Mice IDs are not of relevance here, we exclude the Name column,
#but test all other variables/coefficients/predictors for both df: tube_test_80 and tube_test_40
lm_result_all <- lm(Wins_per_rep ~ .-Name,data=tube_test_80)
summary(lm_result_all)
```

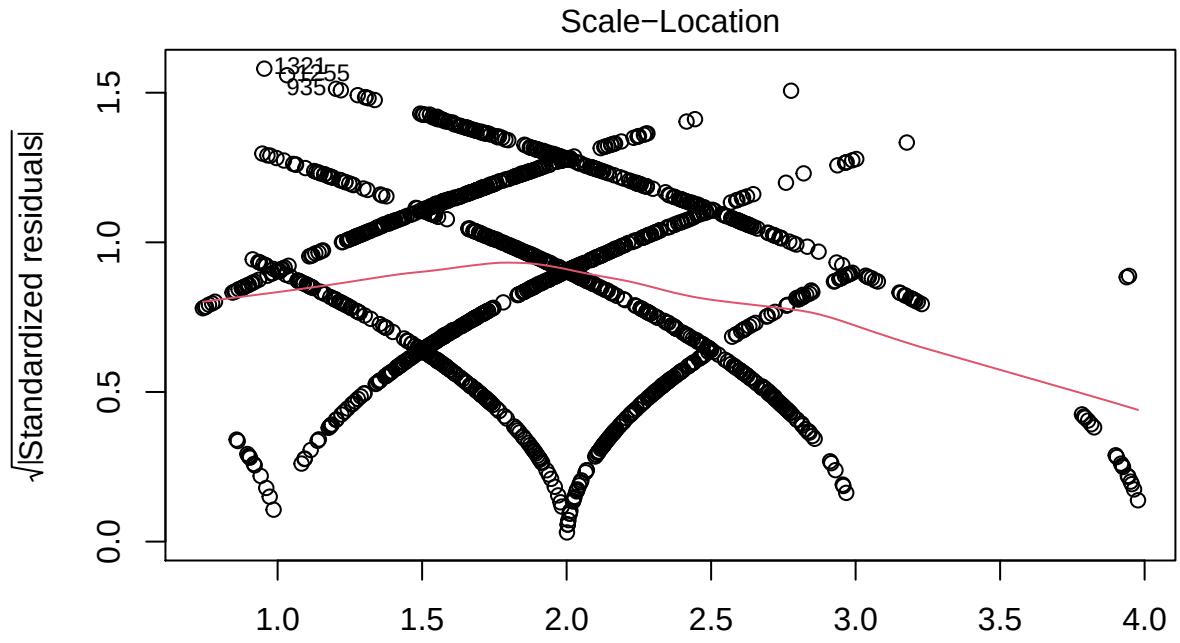
```
##
## Call:
## lm(formula = Wins_per_rep ~ . - Name, data = tube_test_80)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7766 -0.9772 -0.1031  0.9403  3.0461
##
## Coefficients: (10 not defined because of singularities)
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           2.575308   0.427080   6.030 2.08e-09 ***
## TreatmentsHEAT        0.018433   0.110400   0.167 0.867418
```

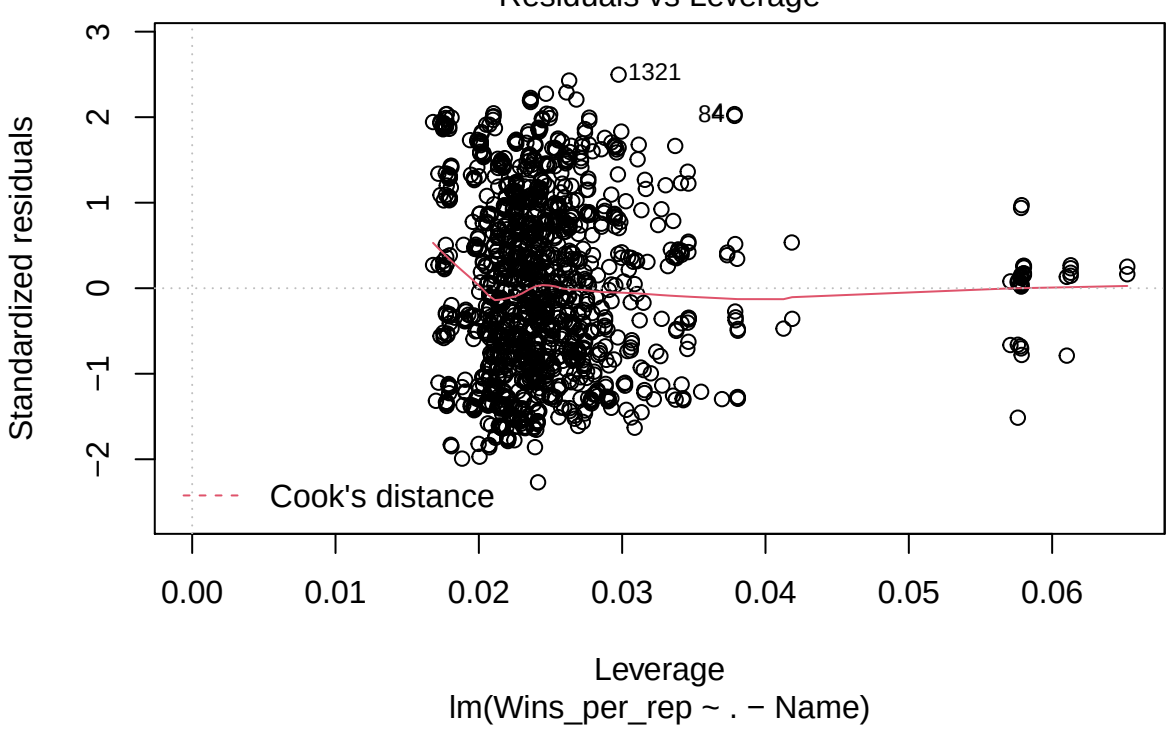
```
## TreatmentsCOLD        0.051767   0.110400   0.469 0.639212
## TreatmentsDIET        0.038120   0.110802   0.344 0.730866
## TreatmentsANTI       -0.109995   0.112421  -0.978 0.328027
## TreatmentsFMT        -0.120770   0.130041  -0.929 0.353194
## ReplicatesR2          0.025381   0.088230   0.288 0.773643
## ReplicatesR3          0.004000   0.086977   0.046 0.963325
## ReplicatesR4          0.046027   0.140104   0.329 0.742566
## ReplicatesR5          0.025013   0.200248   0.125 0.900610
## Arrive_weight         0.002154   0.016420   0.131 0.895642
## PhenotypeBrown       -1.044612   0.104533  -9.993  < 2e-16 ***
## Phenotypecaramel      0.237762   0.340617   0.698 0.485269
## Phenotypedark brown   1.383046   0.337843   4.094 4.48e-05 ***
## PhenotypeLIGHT BROWN -0.472570   0.211802  -2.231 0.025821 *
## PhenotypeWhite       -1.043615   0.108200  -9.645  < 2e-16 ***
## Crate_IDCrate2       -0.370384   0.139157  -2.662 0.007862 **
## Crate_IDCrate3       -0.309293   0.155515  -1.989 0.046908 *
## Crate_IDCrate4       -0.054754   0.240489  -0.228 0.819928
## Crate_IDCrate5        0.269836   0.146165   1.846 0.065081 .
## Crate_IDCrate6        0.262688   0.235866   1.114 0.265586
## Crate_IDCrate7       -0.388380   0.237541  -1.635 0.102266
## Crate_IDCrate8        0.667842   0.246728   2.707 0.006873 **
## SexMale                    NA         NA      NA       NA
## Exp_groupTreatment   -0.313367   0.198050  -1.582 0.113809
## Cage_ID02M            0.722391   0.193997   3.724 0.000204 ***
## Cage_ID03M            0.585602   0.210658   2.780 0.005508 **
## Cage_ID04M            0.264191   0.285637   0.925 0.355162
## Cage_ID05M            0.120988   0.201548   0.600 0.548405
## Cage_ID06M           -0.222318   0.202192  -1.100 0.271717
## Cage_ID07M            0.007626   0.202175   0.038 0.969918
## Cage_ID08M            0.785715   0.216479   3.630 0.000294 ***
## Cage_ID09F           -0.450626   0.214372  -2.102 0.035719 *
## Cage_ID10F            0.743322   0.201076   3.697 0.000227 ***
## Cage_ID11F           -0.054298   0.203884  -0.266 0.790033
## Cage_ID12F            0.361630   0.204612   1.767 0.077372 .
## Cage_ID13F                 NA         NA      NA       NA
## Cage_ID14F            0.455238   0.201021   2.265 0.023683 *
## Cage_ID15F            0.803178   0.205041   3.917 9.38e-05 ***
## Cage_ID16F                 NA         NA      NA       NA
## CrateCrate2                NA         NA      NA       NA
## CrateCrate3                NA         NA      NA       NA
## CrateCrate4                NA         NA      NA       NA
## CrateCrate5                NA         NA      NA       NA
## CrateCrate6                NA         NA      NA       NA
## CrateCrate7                NA         NA      NA       NA
## CrateCrate8                NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.238 on 1450 degrees of freedom
##   (33 observations deleted due to missingness)
## Multiple R-squared:  0.1691, Adjusted R-squared:  0.1485
## F-statistic: 8.197 on 36 and 1450 DF,  p-value: < 2.2e-16
```

```
plot(lm_result_all)
```







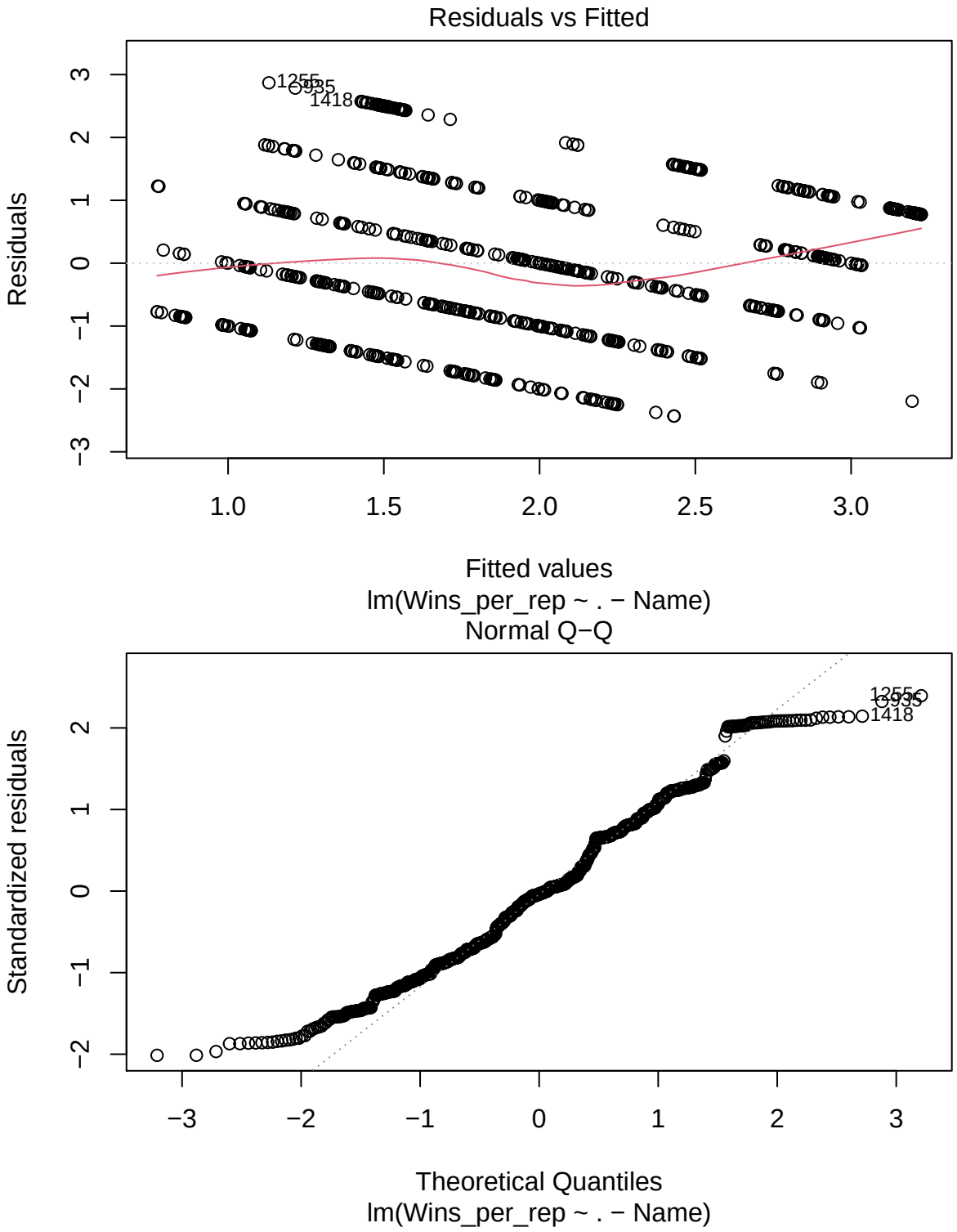


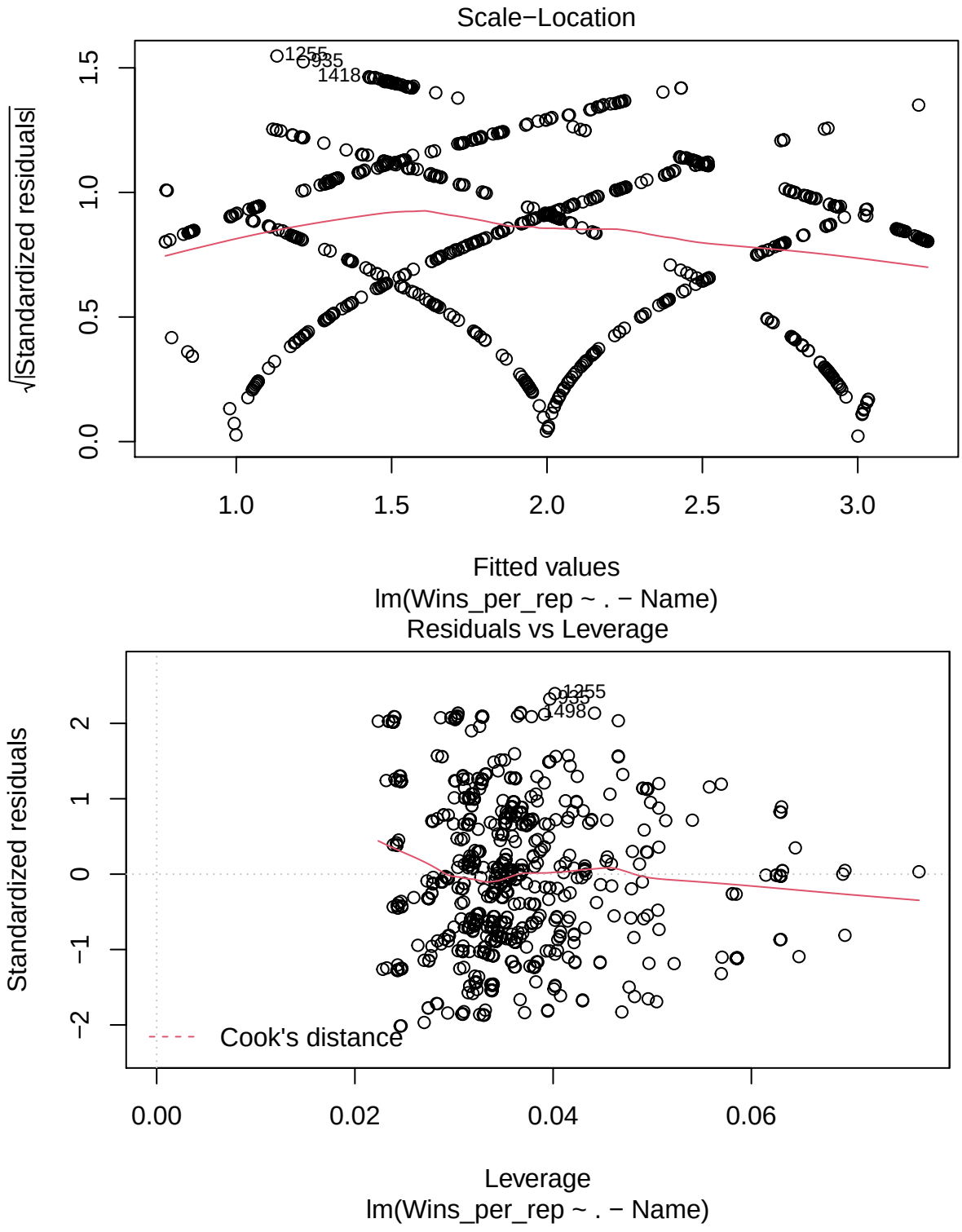

```
lm_result_all <- lm(Wins_per_rep ~ .-Name,data=tube_test_40)
summary(lm_result_all)
```

```
##
## Call:
## lm(formula = Wins_per_rep ~ . - Name, data = tube_test_40)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.43206 -0.95705 -0.04025  0.87291  2.86905
##
## Coefficients: (10 not defined because of singularities)
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           3.618482   0.694919   5.207 2.50e-07 ***
## TreatmentsHEAT        0.014868   0.154077   0.096  0.92315
## TreatmentsCOLD        0.006535   0.154077   0.042  0.96618
## TreatmentsDIET        0.014868   0.154077   0.096  0.92315
## TreatmentsANTI       -0.070838   0.155259  -0.456  0.64834
## TreatmentsFMT        -0.058325   0.179371  -0.325  0.74515
## ReplicatesR2         -0.005025   0.122585  -0.041  0.96731
## ReplicatesR3         -0.003941   0.120822  -0.033  0.97399
## ReplicatesR4         -0.018515   0.194456  -0.095  0.92417
## ReplicatesR5          0.001593   0.275172   0.006  0.99538
## Arrive_weight        -0.023974   0.026451  -0.906  0.36504
## PhenotypeBrown       -1.331121   0.139145  -9.566  < 2e-16 ***
## PhenotypeLIGHT BROWN  0.217656   0.341217   0.638  0.52375
## PhenotypeWhite       -1.439426   0.151814  -9.482  < 2e-16 ***
## Crate_IDCrate2        0.561053   0.193550   2.899  0.00386 **
## Crate_IDCrate3       -0.284023   0.269166  -1.055  0.29169
## Crate_IDCrate4       -0.067226   0.337371  -0.199  0.84211
## Crate_IDCrate5        0.484845   0.175304   2.766  0.00582 **
## Crate_IDCrate6       -0.309033   0.291899  -1.059  0.29009
## Crate_IDCrate7       -0.703396   0.310029  -2.269  0.02357 *
## Crate_IDCrate8        0.262489   0.365814   0.718  0.47327
## SexMale                     NA         NA      NA       NA
## Exp_groupTreatment   -0.126818   0.218510  -0.580  0.56184
## Cage_ID04M           -0.431109   0.296289  -1.455  0.14609
## Cage_ID05M           -0.714613   0.239672  -2.982  0.00296 **
## Cage_ID06M           -0.994507   0.222106  -4.478 8.76e-06 ***
## Cage_ID10F            0.518879   0.202433   2.563  0.01057 *
## Cage_ID11F           -0.452999   0.235648  -1.922  0.05495 .
## Cage_ID12F                  NA         NA      NA       NA
## Cage_ID13F                  NA         NA      NA       NA
## CrateCrate2                 NA         NA      NA       NA
## CrateCrate3                 NA         NA      NA       NA
## CrateCrate4                 NA         NA      NA       NA
## CrateCrate5                 NA         NA      NA       NA
## CrateCrate6                 NA         NA      NA       NA
## CrateCrate7                 NA         NA      NA       NA
## CrateCrate8                 NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.223 on 727 degrees of freedom
##   (6 observations deleted due to missingness)
## Multiple R-squared:  0.2183, Adjusted R-squared:  0.1903
## F-statistic: 7.807 on 26 and 727 DF,  p-value: < 2.2e-16
```

```
plot(lm_result_all)
```


Residuals vs Fitted
1255
3935
1418
Residuals
3
2
1
0
−1
−2
−3
1.0
1.5
2.0
2.5
3.0
Fitted values
lm(Wins_per_rep ~ . − Name)
Normal Q−Q
1255
3935
1418
Standardized residuals
2
1
0
−1
−2
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(Wins_per_rep ~ . − Name)



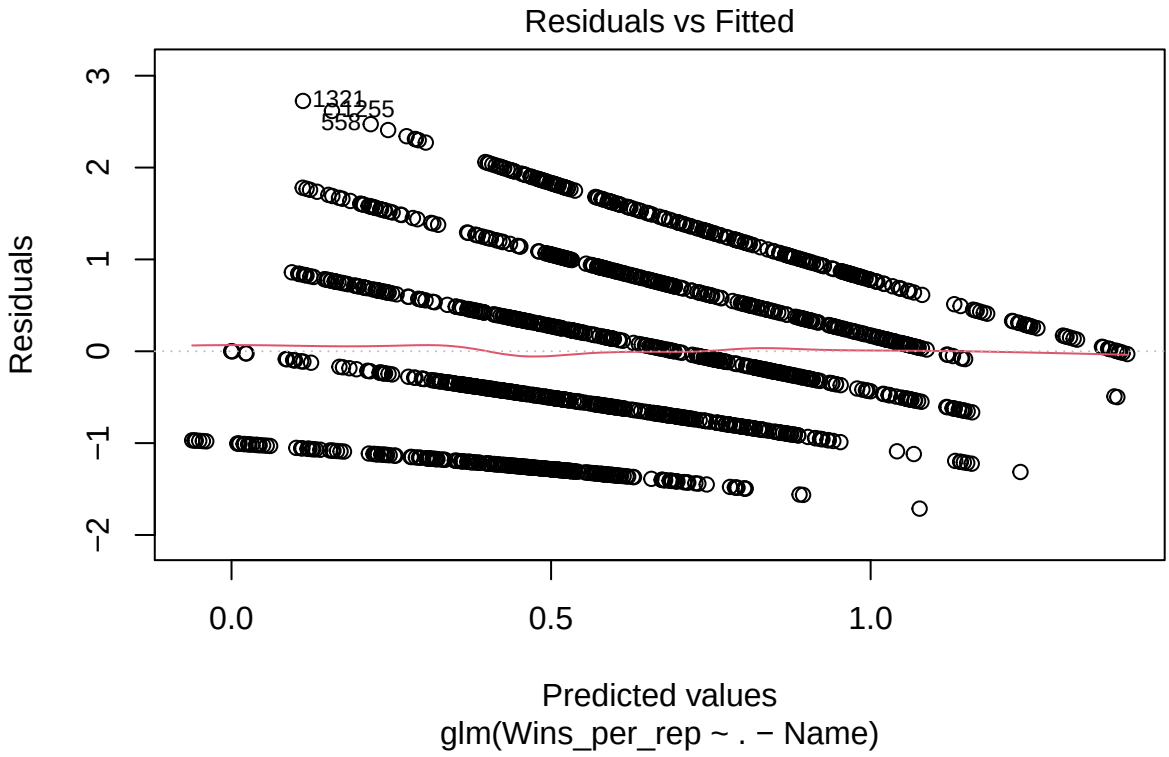
Here I test the data using glm()

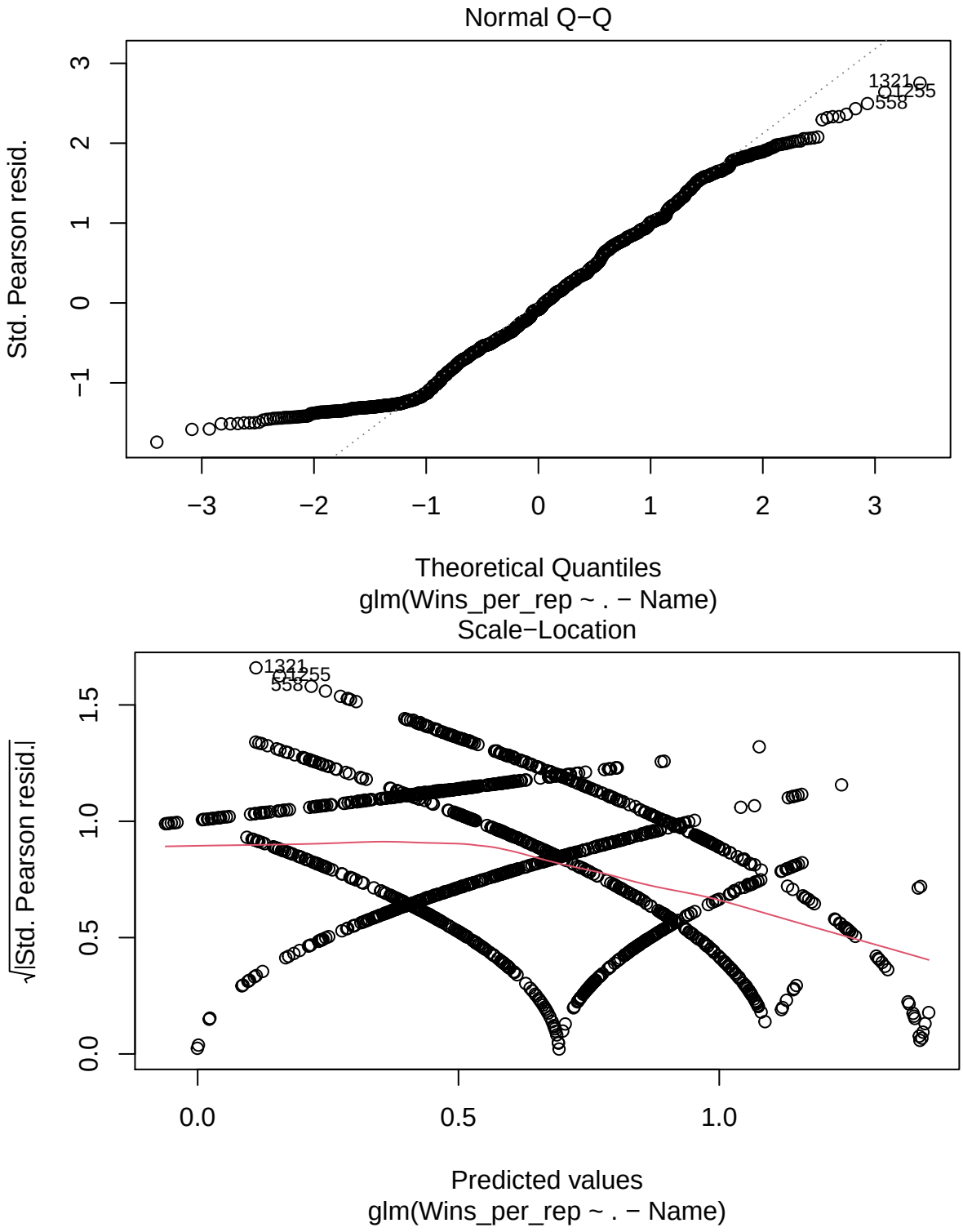
```
#Because the Mice IDs are not of relevance here, we exclude the Name column,
#but test all other variables/coefficients/predictors for both df: tube_test_80 and tube_test_40
glm_result <- glm(Wins_per_rep ~ .-Name,data=tube_test_80,family = poisson())
summary(glm_result)
```

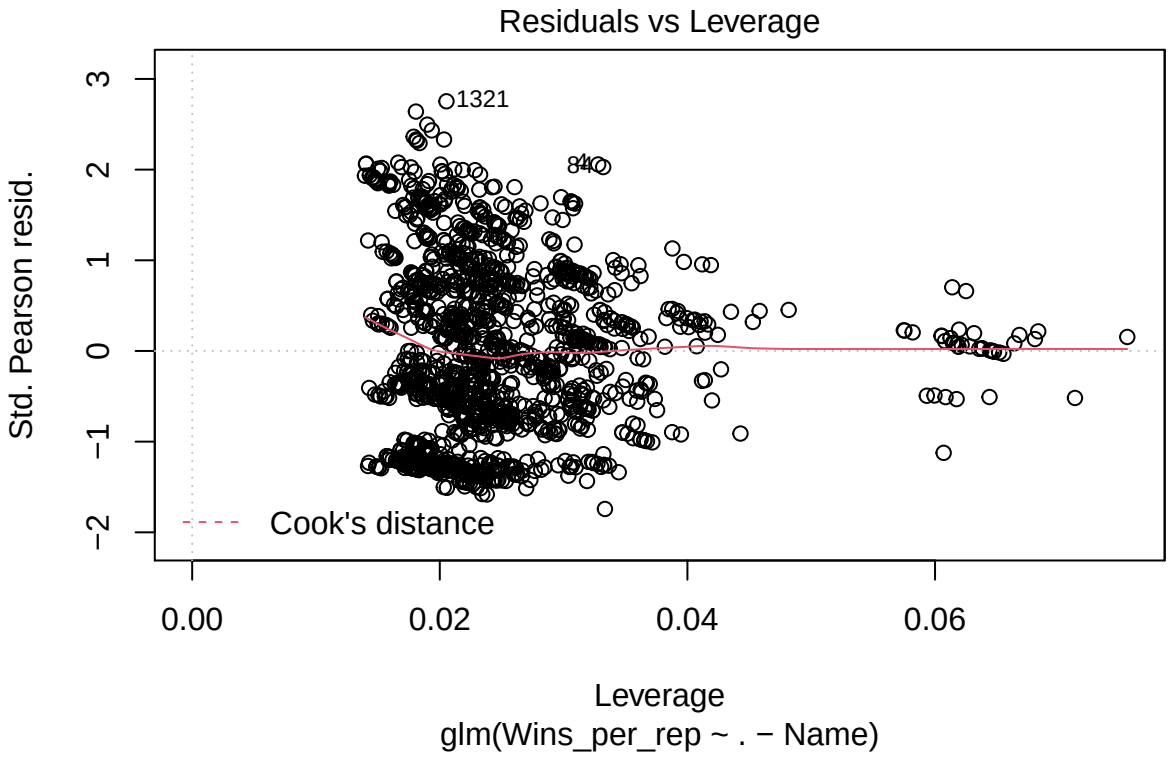
```
##
## Call:
## glm(formula = Wins_per_rep ~ . - Name, family = poisson(), data = tube_test_80)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.42271  -0.77571  -0.08021   0.65960   2.10534
##
## Coefficients: (10 not defined because of singularities)
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)             0.985206   0.250896   3.927 8.61e-05 ***
## TreatmentsHEAT          0.009598   0.064018   0.150 0.880826
## TreatmentsCOLD          0.026547   0.063737   0.417 0.677035
## TreatmentsDIET          0.019482   0.064033   0.304 0.760933
## TreatmentsANTI         -0.057376   0.066122  -0.868 0.385540
## TreatmentsFMT          -0.063008   0.076455  -0.824 0.409875
## ReplicatesR2            0.013072   0.051132   0.256 0.798218
## ReplicatesR3            0.002011   0.050561   0.040 0.968277
## ReplicatesR4            0.024022   0.081529   0.295 0.768262
## ReplicatesR5            0.013077   0.118303   0.111 0.911981
## Arrive_weight           0.001826   0.010085   0.181 0.856304
## PhenotypeBrown         -0.542872   0.060904  -8.914  < 2e-16 ***
## Phenotypecaramel        0.124981   0.177132   0.706 0.480448
## Phenotypedark brown     0.445839   0.162741   2.740 0.006152 **
## PhenotypeLIGHT BROWN -0.220393   0.123177  -1.789 0.073577 .
## PhenotypeWhite         -0.538552   0.064836  -8.306  < 2e-16 ***
## Crate_IDCrate2         -0.202961   0.084941  -2.389 0.016875 *
## Crate_IDCrate3         -0.177914   0.093150  -1.910 0.056136 .
## Crate_IDCrate4         -0.051602   0.149047  -0.346 0.729180
## Crate_IDCrate5          0.110287   0.084200   1.310 0.190259
## Crate_IDCrate6          0.103837   0.141628   0.733 0.463459
## Crate_IDCrate7         -0.254371   0.149675  -1.699 0.089228 .
## Crate_IDCrate8          0.310743   0.149235   2.082 0.037321 *
## SexMale                       NA         NA      NA       NA
## Exp_groupTreatment     -0.213216   0.122260  -1.744 0.081167 .
## Cage_ID02M              0.403339   0.117941   3.420 0.000627 ***
## Cage_ID03M              0.352764   0.131309   2.687 0.007220 **
## Cage_ID04M              0.089943   0.174109   0.517 0.605445
## Cage_ID05M              0.078818   0.121094   0.651 0.515120
## Cage_ID06M             -0.071373   0.121449  -0.588 0.556747
## Cage_ID07M              0.024071   0.125045   0.192 0.847354
## Cage_ID08M              0.449968   0.132245   3.403 0.000668 ***
## Cage_ID09F             -0.198612   0.137774  -1.442 0.149422
## Cage_ID10F              0.451013   0.124131   3.633 0.000280 ***
## Cage_ID11F              0.005785   0.124976   0.046 0.963078
## Cage_ID12F              0.209845   0.125971   1.666 0.095748 .
## Cage_ID13F                    NA         NA      NA       NA
## Cage_ID14F              0.298496   0.123355   2.420 0.015528 *
## Cage_ID15F              0.480633   0.127573   3.767 0.000165 ***
## Cage_ID16F                    NA         NA      NA       NA
## CrateCrate2                   NA         NA      NA       NA
## CrateCrate3                   NA         NA      NA       NA
## CrateCrate4                   NA         NA      NA       NA
## CrateCrate5                   NA         NA      NA       NA
```

```
## CrateCrate6                  NA         NA      NA       NA
## CrateCrate7                  NA         NA      NA       NA
## CrateCrate8                  NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 1787.1  on 1486  degrees of freedom
## Residual deviance: 1560.5  on 1450  degrees of freedom
##   (33 observations deleted due to missingness)
## AIC: 4872.6
##
## Number of Fisher Scoring iterations: 5
```

```
plot(glm_result)
```




Normal Q−Q
Std. Pearson resid.
1321
1255
558
−3
−2
−1
0
1
2
3
Theoretical Quantiles
glm(Wins_per_rep ~ . − Name)
Scale−Location
√|Std. Pearson resid.|
0.0
0.5
1.0
1.5
Predicted values
glm(Wins_per_rep ~ . − Name)

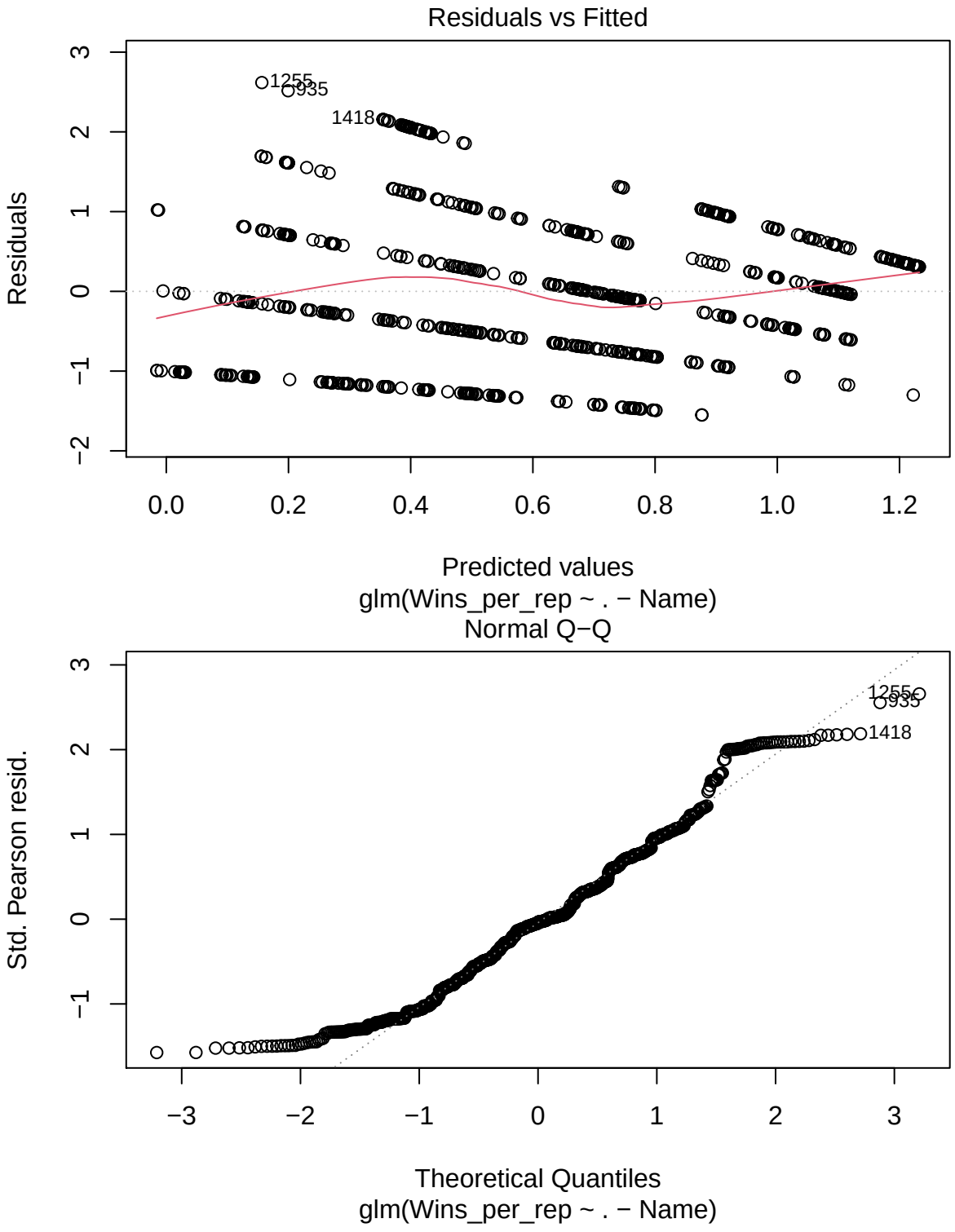


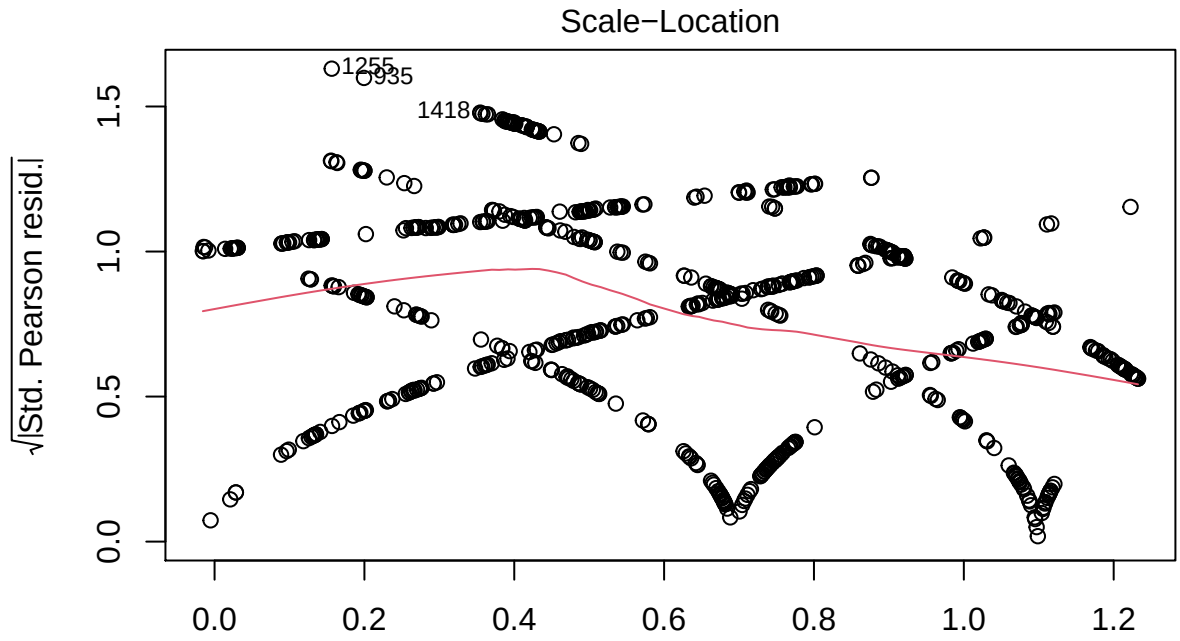

```
glm_result <- glm(Wins_per_rep ~ .-Name,data=tube_test_40,family = poisson())
summary(glm_result)
```

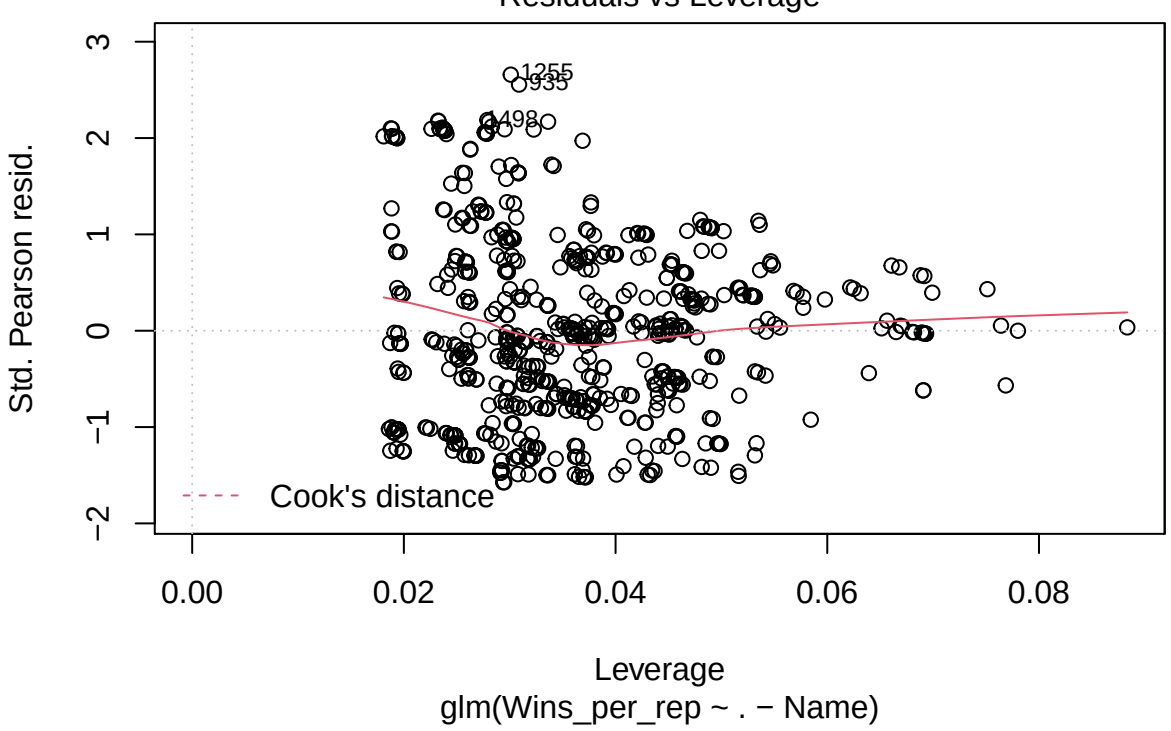
```
##
## Call:
## glm(formula = Wins_per_rep ~ . - Name, family = poisson(), data = tube_test_40)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.19239  -0.77496  -0.03654   0.58023   2.04397
##
## Coefficients: (10 not defined because of singularities)
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)             1.2880107  0.4207857   3.061 0.002206 **
## TreatmentsHEAT          0.0074473  0.0892944   0.083 0.933532
## TreatmentsCOLD          0.0032720  0.0893920   0.037 0.970802
## TreatmentsDIET          0.0074473  0.0892944   0.083 0.933532
## TreatmentsANTI         -0.0361338  0.0907774  -0.398 0.690595
## TreatmentsFMT          -0.0297379  0.1049425  -0.283 0.776891
## ReplicatesR2           -0.0025349  0.0712018  -0.036 0.971601
## ReplicatesR3           -0.0019785  0.0701821  -0.028 0.977510
## ReplicatesR4           -0.0094148  0.1134252  -0.083 0.933848
## ReplicatesR5            0.0009109  0.1610300   0.006 0.995486
## Arrive_weight          -0.0036823  0.0164730  -0.224 0.823119
## PhenotypeBrown         -0.6409863  0.0782605  -8.190 2.60e-16 ***
## PhenotypeLIGHT BROWN    0.0787051  0.1718906   0.458 0.647039
## PhenotypeWhite         -0.7301891  0.0921219  -7.926 2.26e-15 ***
## Crate_IDCrate2          0.3247412  0.1136101   2.858 0.004258 **
## Crate_IDCrate3         -0.1621914  0.1753692  -0.925 0.355040
```

```
## Crate_IDCrate4       -0.0275113  0.2087908  -0.132 0.895170
## Crate_IDCrate5        0.2385385  0.1055724   2.259 0.023854 *
## Crate_IDCrate6       -0.1391767  0.1749053  -0.796 0.426191
## Crate_IDCrate7       -0.3895277  0.1966837  -1.980 0.047650 *
## Crate_IDCrate8        0.1731069  0.2250214   0.769 0.441721
## SexMale                      NA         NA      NA       NA
## Exp_groupTreatment   -0.0607722  0.1359041  -0.447 0.654753
## Cage_ID04M           -0.2291182  0.1803810  -1.270 0.204017
## Cage_ID05M           -0.3984480  0.1445578  -2.756 0.005846 **
## Cage_ID06M           -0.4778446  0.1314431  -3.635 0.000278 ***
## Cage_ID10F            0.2945573  0.1264076   2.330 0.019795 *
## Cage_ID11F           -0.2249200  0.1489704  -1.510 0.131087
## Cage_ID12F                   NA         NA      NA       NA
## Cage_ID13F                   NA         NA      NA       NA
## CrateCrate2                  NA         NA      NA       NA
## CrateCrate3                  NA         NA      NA       NA
## CrateCrate4                  NA         NA      NA       NA
## CrateCrate5                  NA         NA      NA       NA
## CrateCrate6                  NA         NA      NA       NA
## CrateCrate7                  NA         NA      NA       NA
## CrateCrate8                  NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 906.16  on 753  degrees of freedom
## Residual deviance: 753.71  on 727  degrees of freedom
##   (6 observations deleted due to missingness)
## AIC: 2471.5
##
## Number of Fisher Scoring iterations: 5
```

```
plot(glm_result)
```


Residuals vs Fitted
Residuals
Predicted values
glm(Wins_per_rep ~ . − Name)
1255
935
1418
Normal Q−Q
Std. Pearson resid.
Theoretical Quantiles
glm(Wins_per_rep ~ . − Name)





I read about testing families and testing log transformed data using glm(). I tried to test different things, but not all codes work for me. If you Inkaki find this relevant, then I would like some help.

```
#However, other families may be a better fit for my data, and thus, I read about them using ?glm()
#I also used ChatGPT to help me understand the definition of each family
#Turns out the Quasipoisson could fit my data better than poisson
```

```r
#So to find out if this is the case, I have to explore my data
#If the  variance is larger than the mean in my dataset (=overdispersion), then quasipoisson is a bette
#You only calculate the variance (sd) and mean of the dependent variable, which in my case is number of

#First testing this with all 80 mice
dependent_variable_wins <- tube_test_80$Wins_per_rep
mean_value <- mean(dependent_variable_wins,na.rm=TRUE)
sd_value <- sd(dependent_variable_wins,na.rm=TRUE)

coefficient_dispersion <- (sd_value / mean_value) * 100
coefficient_variation <- (sd_value / mean_value) * 100

print(coefficient_dispersion)
```

```
## [1] 69.62755
```

```r
print(coefficient_variation)
```

```
## [1] 69.62755
```

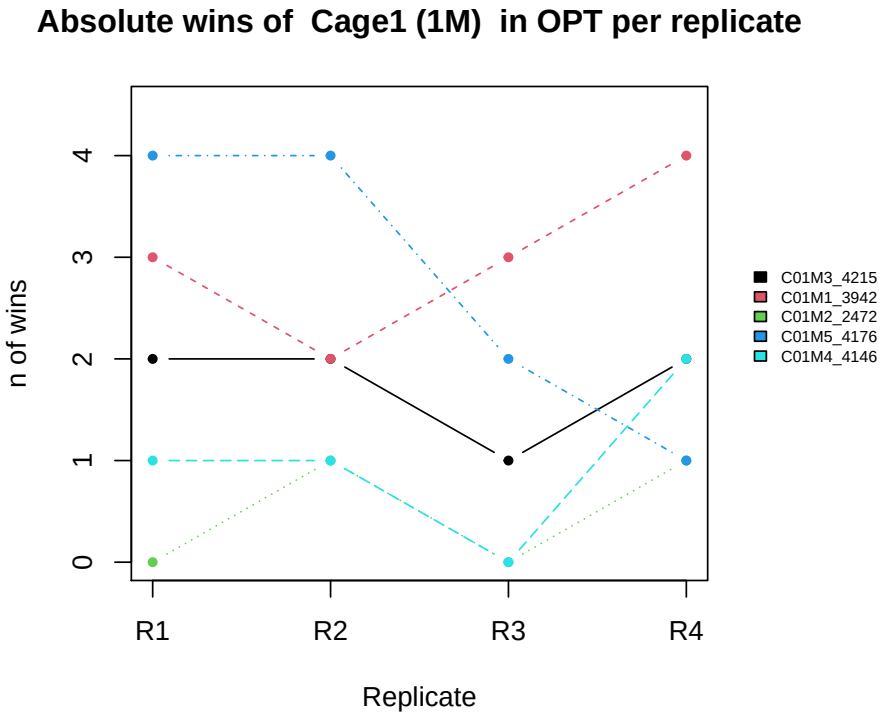
```r
#The results of the 40 specific mice (tube_test_40) were: 68.8126 and 68.8126
#In this case, the standard deviation (sd) = variance is equal to the mean, resulting in the coefficien
#So since my data is count data with no overdispersion, the Poisson family might be suitable indeed.
#When the standard deviation is equal to the mean, it implies that the data values are evenly distribut
#This can occur when the data is symmetrically distributed or when there is a specific relationship bet

#I looked at papers doing tube test analyses and one named "social hierarchy position in female mice is
#I want to see if the Poisson-lognormal distribution test fits my data better than regular poisson

#TEST ON SPECIFIC 40 MICE
# Fit Poisson regression model
#poisson_model <- glm(Wins_per_rep ~ . - Name, data = cleaned_tube_test_40, family = "poisson")
poisson_model <- glm(Wins_per_rep ~ Treatments + Cage_ID, data = cleaned_tube_test_40, family = "poisso

# Fit Poisson-lognormal regression model with log-transformed response
#log_count_var <- log(cleaned_tube_test_40$Wins_per_rep)
#cleaned_tube_test_40$log_wins <- log_count_var
#head(cleaned_tube_test_40)
#poisson_lognormal_model <- glm(log_wins ~ . - Name, data = cleaned_tube_test_40, family = "gaussian") 
#poisson_lognormal_model <- glm(log_wins ~ Treatments + Cage_ID, data = cleaned_tube_test_40, family = 
#cleaned_tube_test_40 <- within(cleaned_tube_test_40, rm(log_wins))
#poisson_lognormal_model <- glm(log(Wins_per_rep) ~ . - Name, data = cleaned_tube_test_40, family = "ga
#poisson_lognormal_model <- glm(log(Wins_per_rep) ~ Treatments + Cage_ID, data = cleaned_tube_test_40, 
```

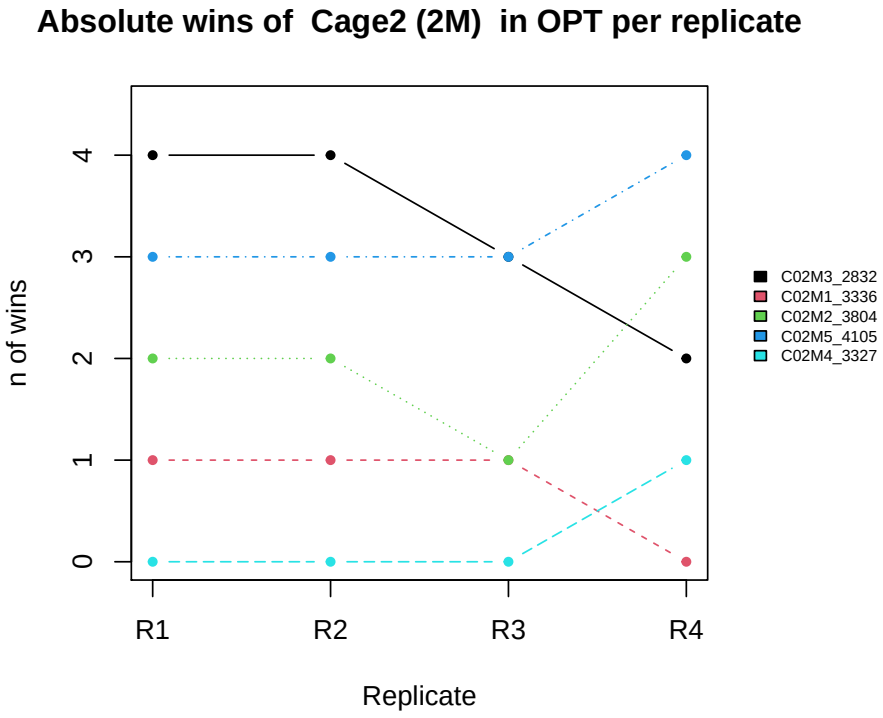
Visualizations based on simple questions

Antton and Ostaizka recommened me to ask simple questions and try to answer those via visualizations. They are in this section asked and then the associated plots are my attempts of answering them. I showed you some of them them in the presentation end August. I would love to get your inputs Inaki; if you think I answer the questions by choosing the plots I did, or if you think I should visualize the data differently.

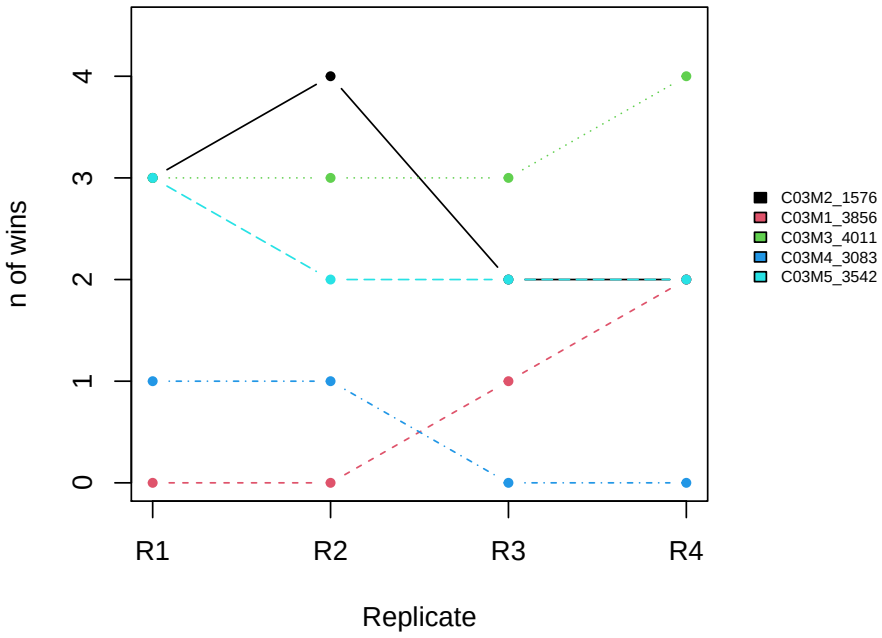
FIRST QUESTION: 1. What is the dominance hierarchy per cage for each treatments? Is the hierarchy stable/the same across tube test replicates?

I try to answer this by visualizing the absolute number of wins per mouse per replicate for the OPT and FMT treatments. I have done similar plots of the other treatment, but do not show them here. If you Inaki want me to, then please let me know.

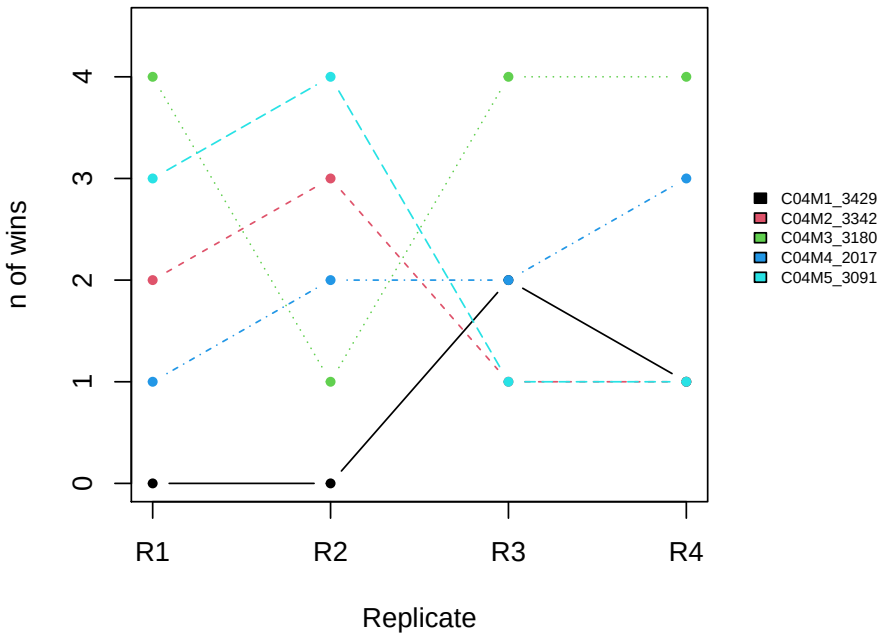




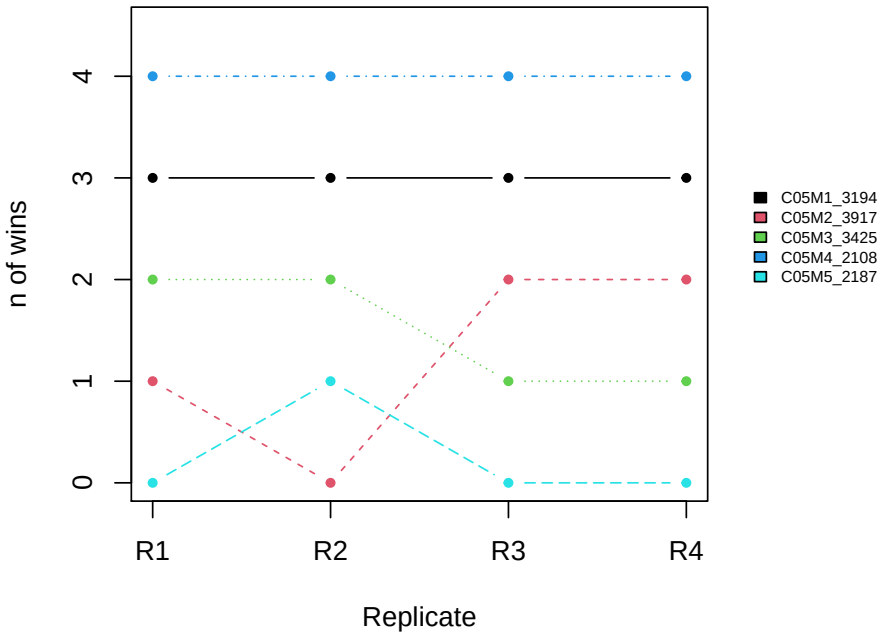
Absolute wins of Cage3 (3M) in OPT per replicate



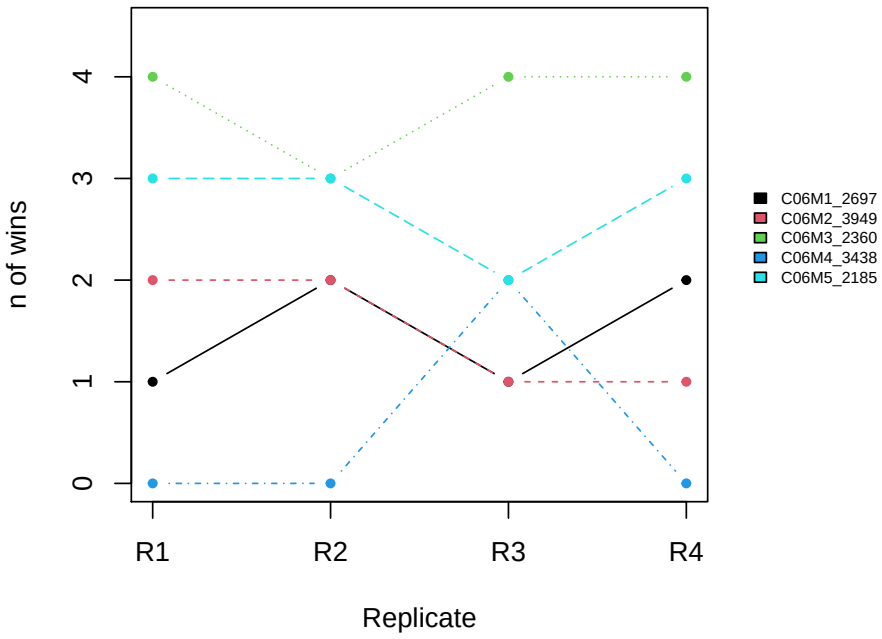
Absolute wins of Cage4 (4M) in OPT per replicate



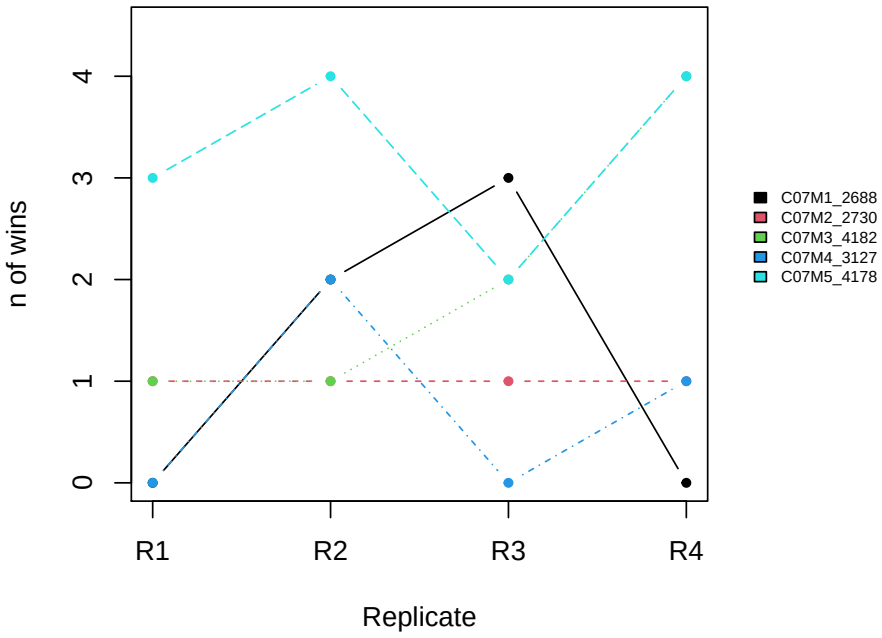
Absolute wins of Cage5 (5M) in OPT per replicate



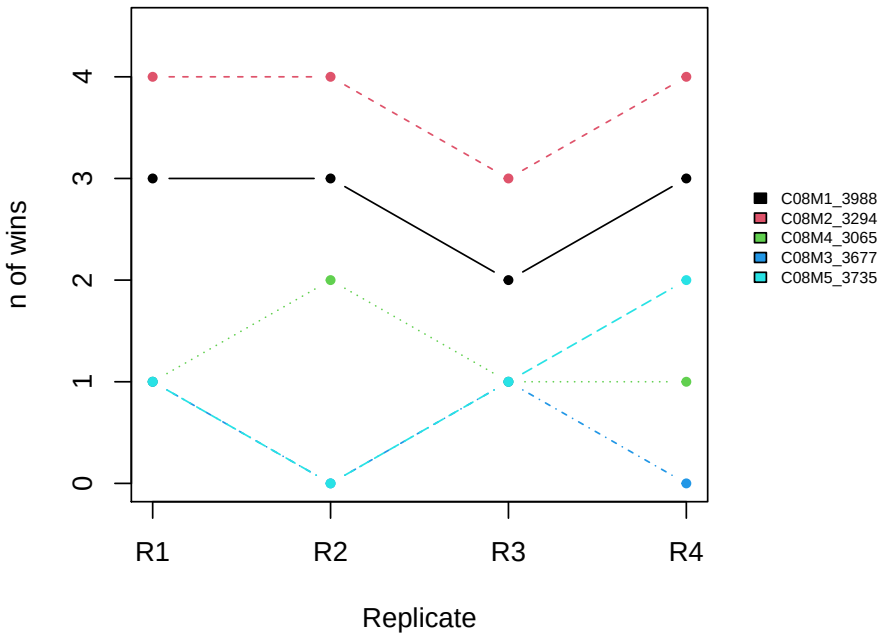
Absolute wins of Cage6 (6M) in OPT per replicate



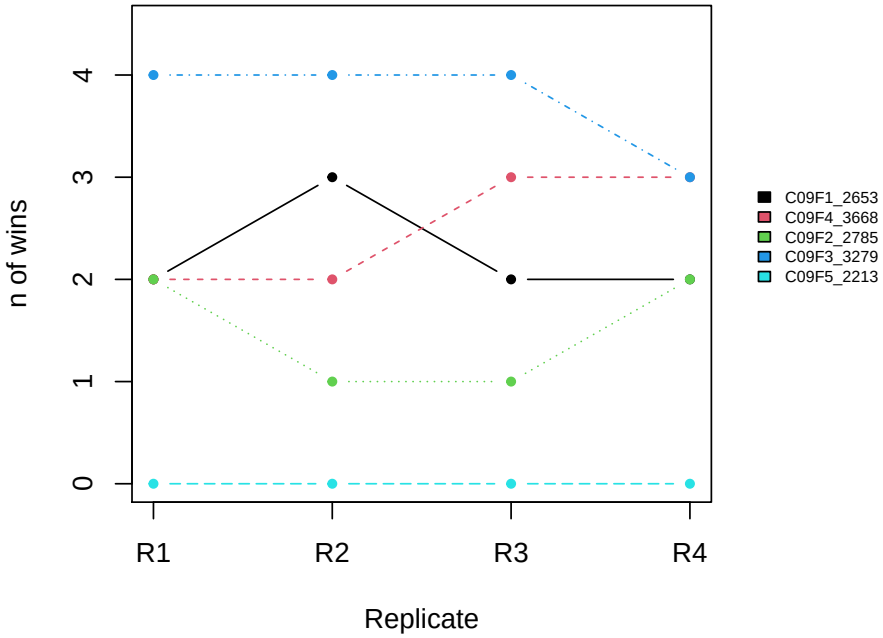
Absolute wins of Cage7 (7M) in OPT per replicate



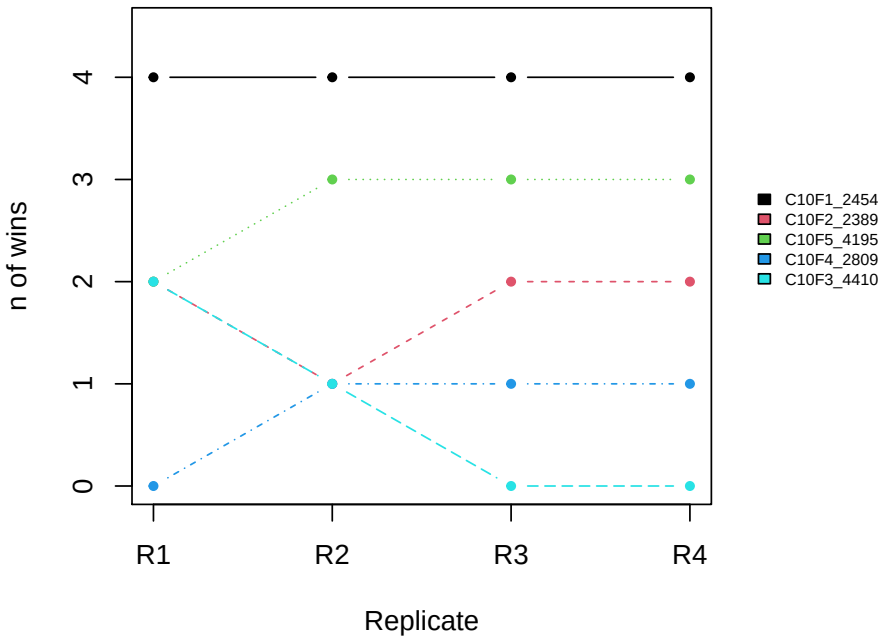
Absolute wins of Cage8 (8M) in OPT per replicate



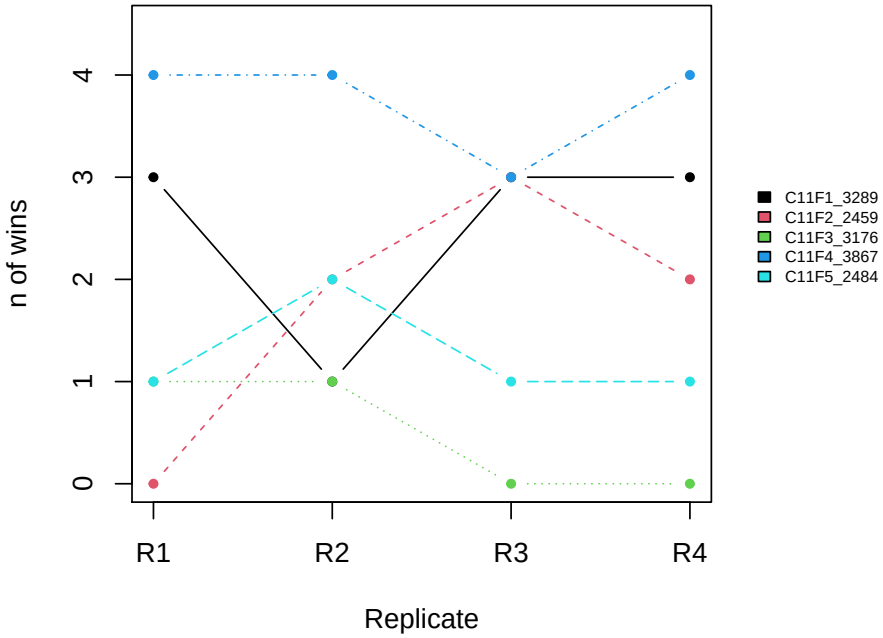
Absolute wins of Cage9 (9F) in OPT per replicate



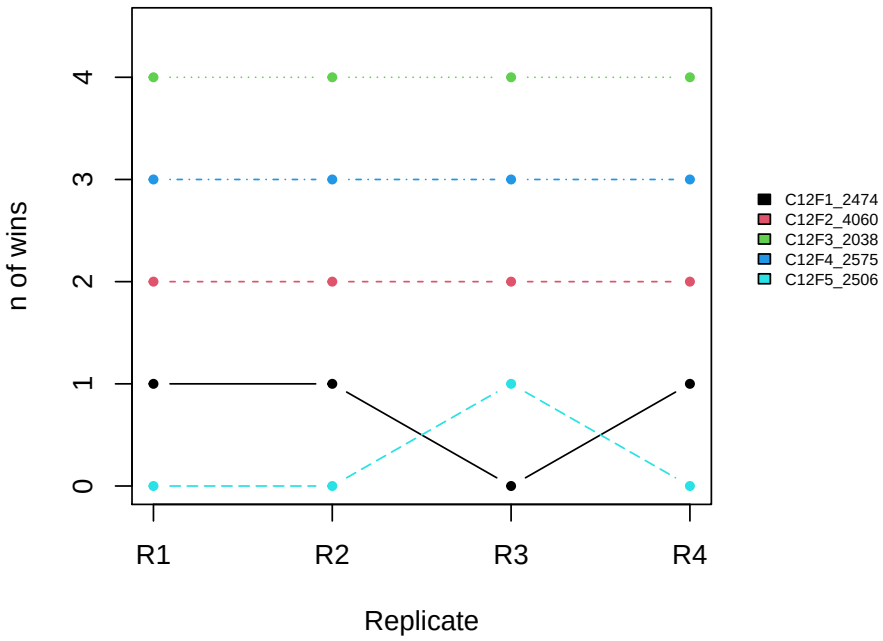
Absolute wins of Cage10 (10F) in OPT per replicate



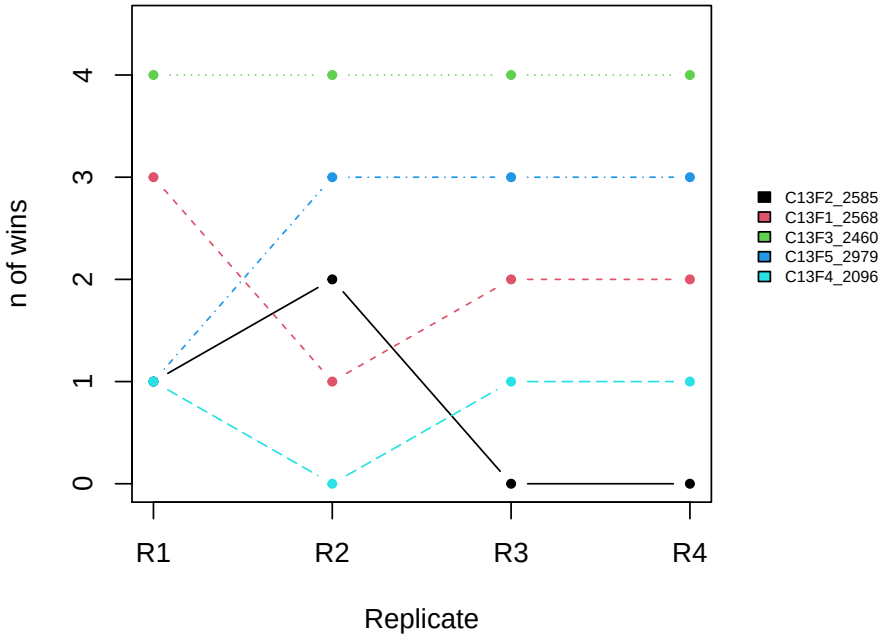
Absolute wins of Cage11 (11F) in OPT per replicate



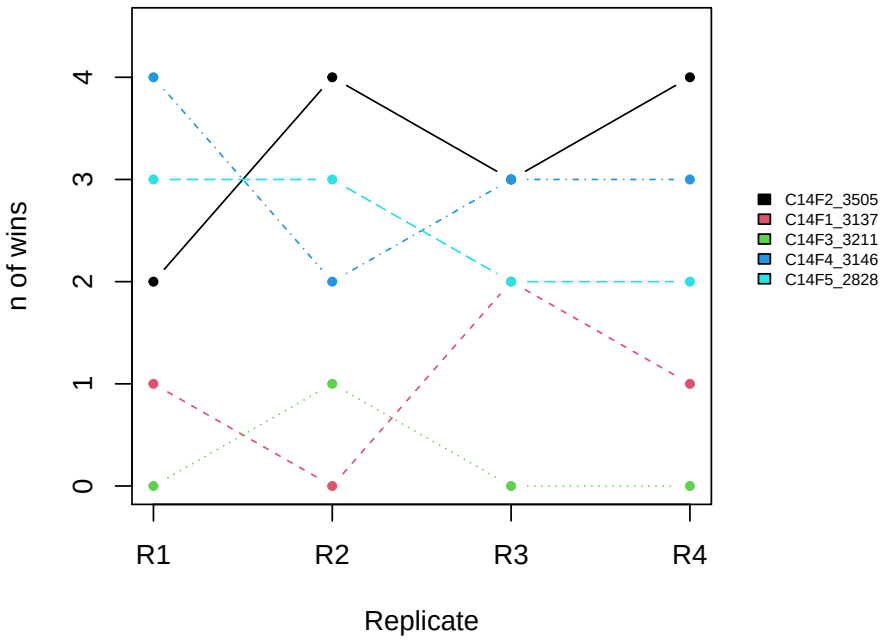
Absolute wins of Cage12 (12F) in OPT per replicate



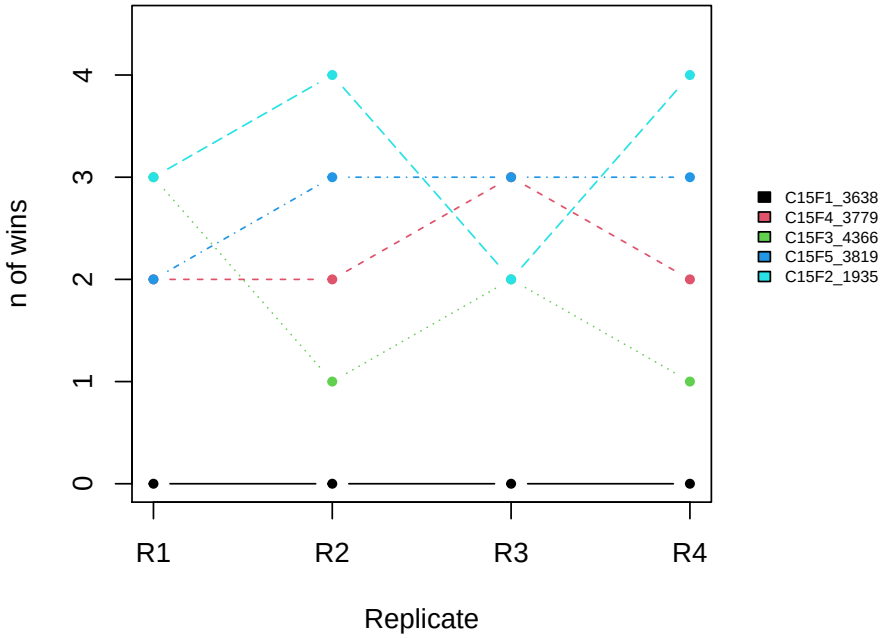
Absolute wins of Cage13 (13F) in OPT per replicate



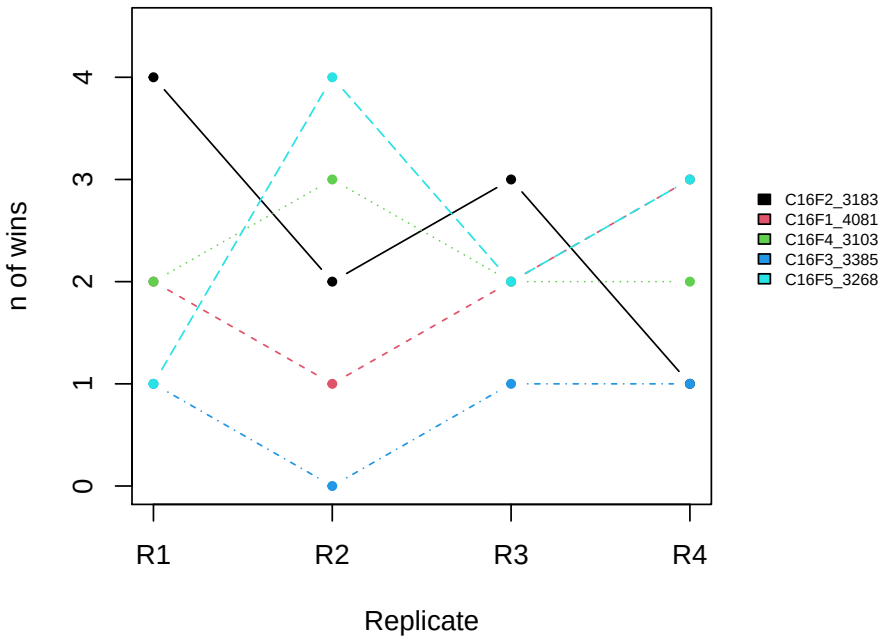
Absolute wins of Cage14 (14F) in OPT per replicate



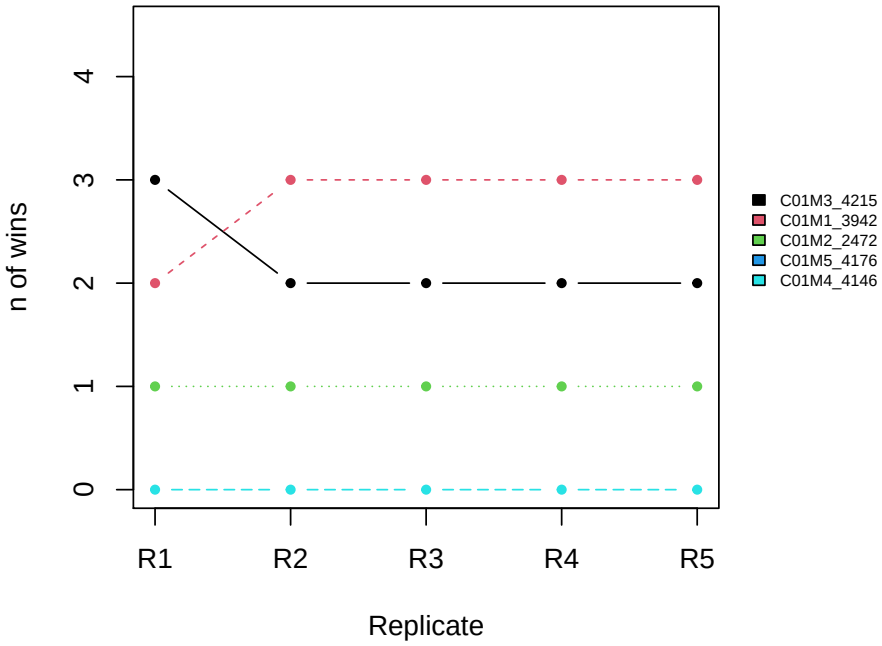
Absolute wins of Cage15 (15F) in OPT per replicate



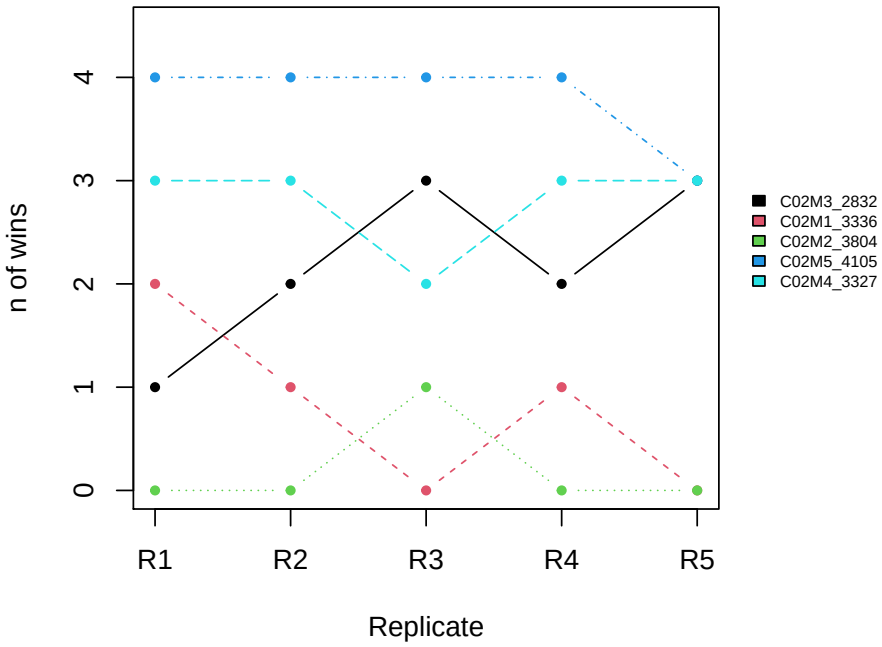
Absolute wins of Cage16 (16F) in OPT per replicate



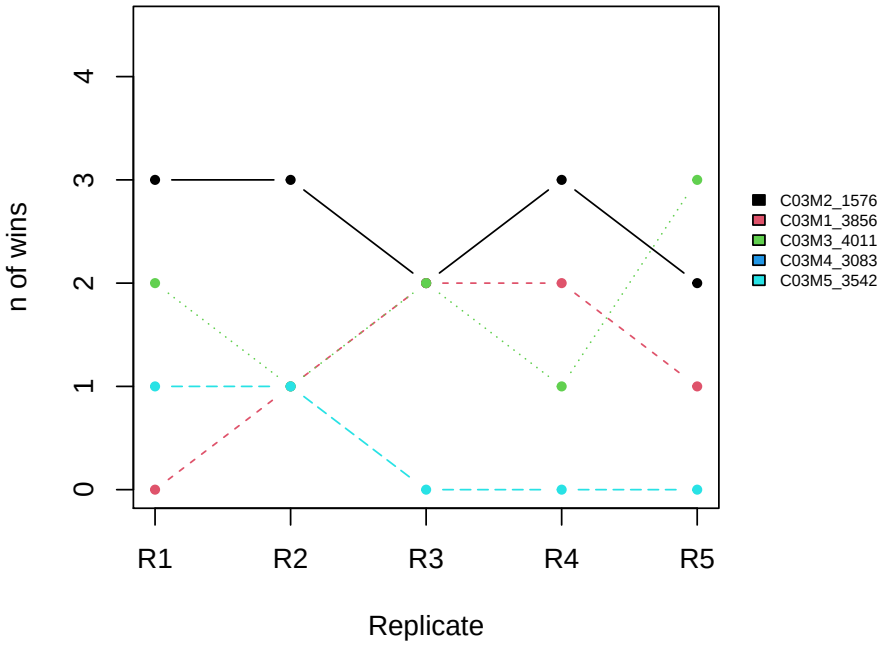
Absolute wins of Cage1 (1M) in FMT per replicate



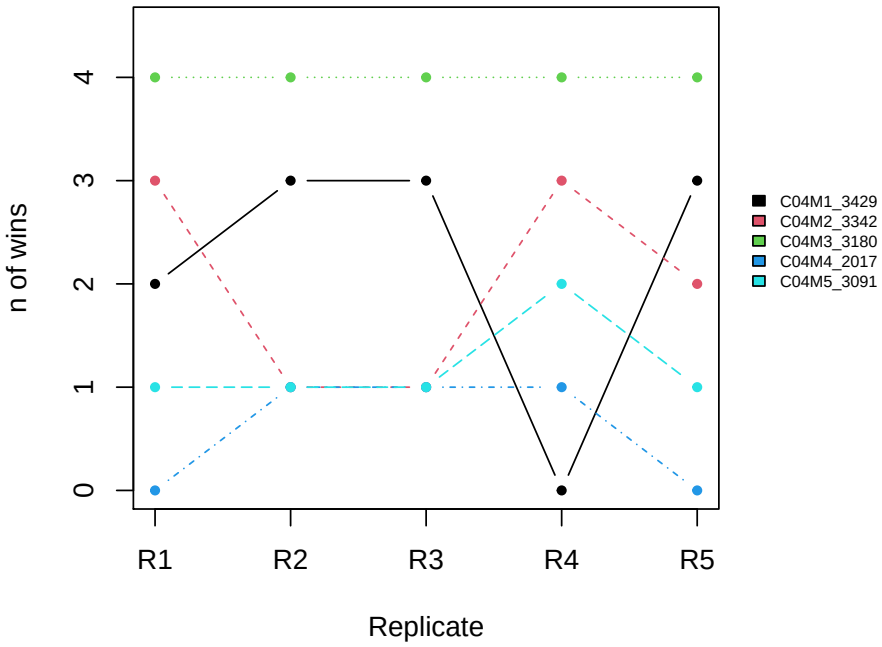
Absolute wins of Cage2 (2M) in FMT per replicate



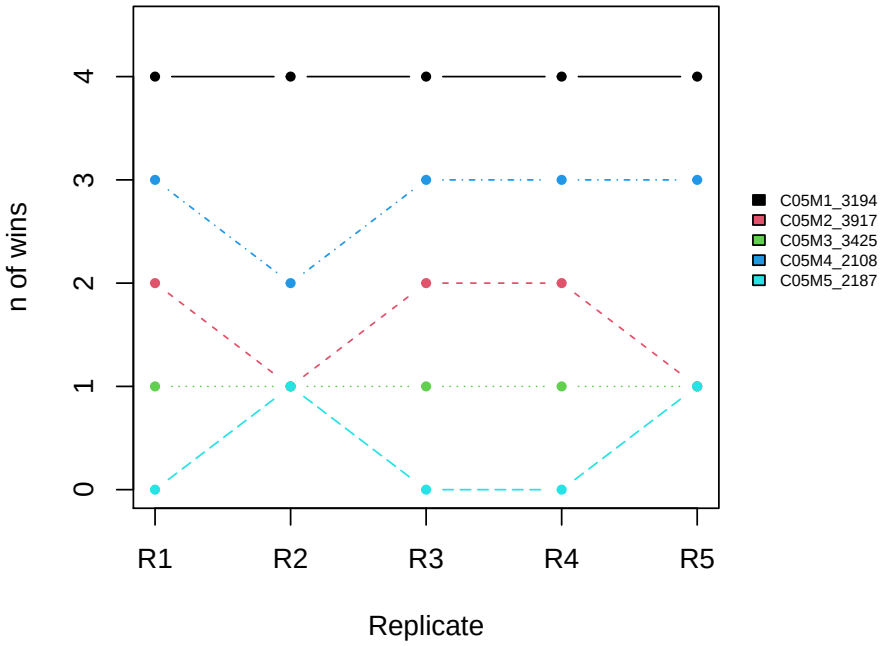
Absolute wins of Cage3 (3M) in FMT per replicate



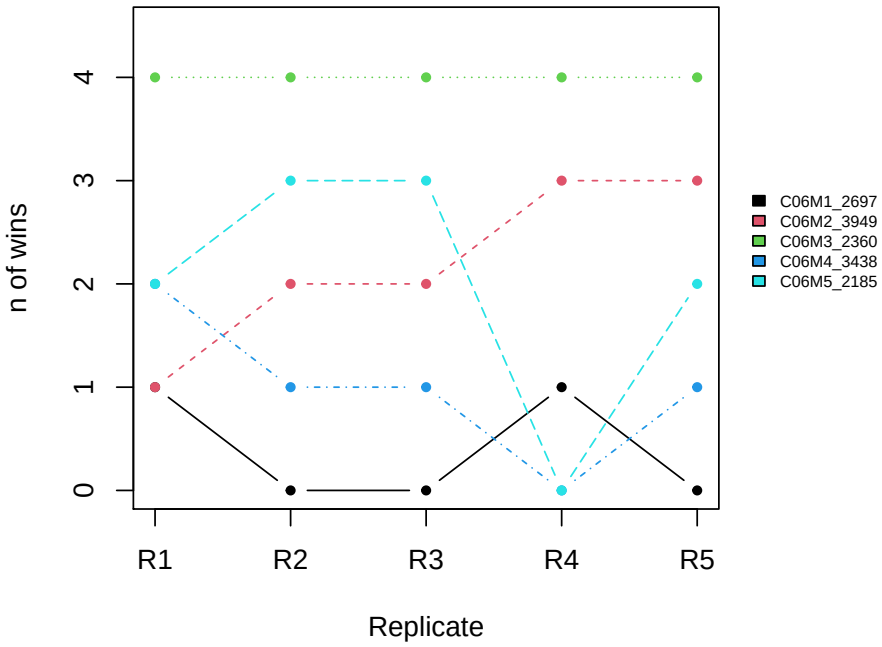
Absolute wins of Cage4 (4M) in FMT per replicate

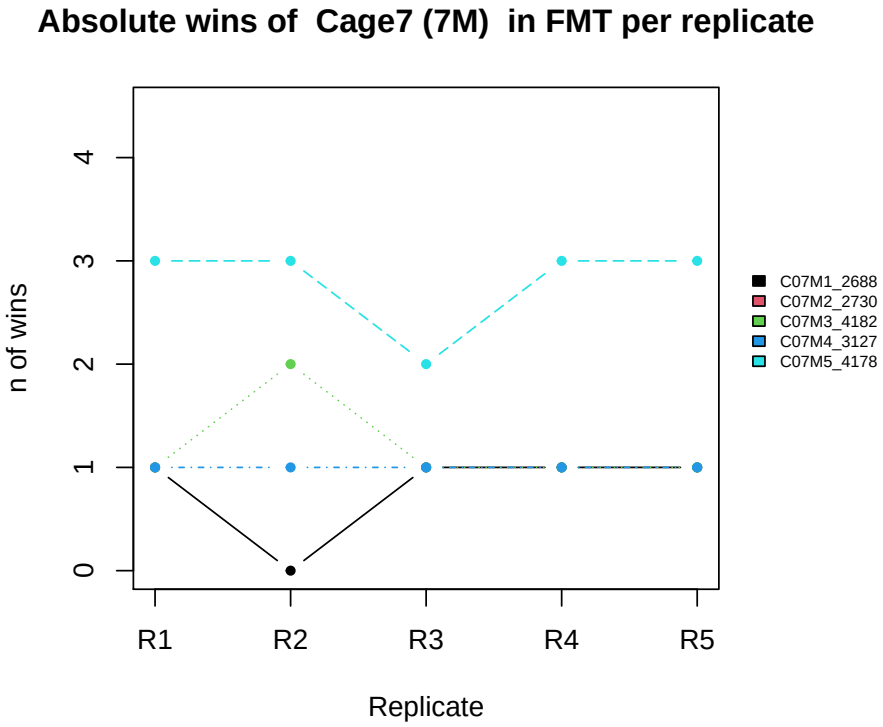


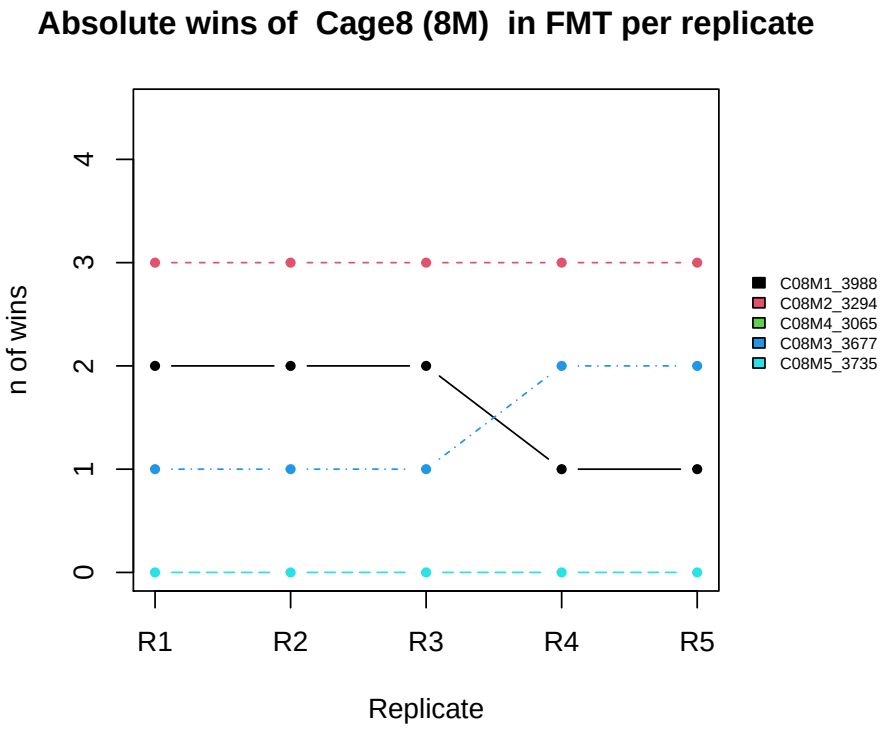
Absolute wins of Cage5 (5M) in FMT per replicate



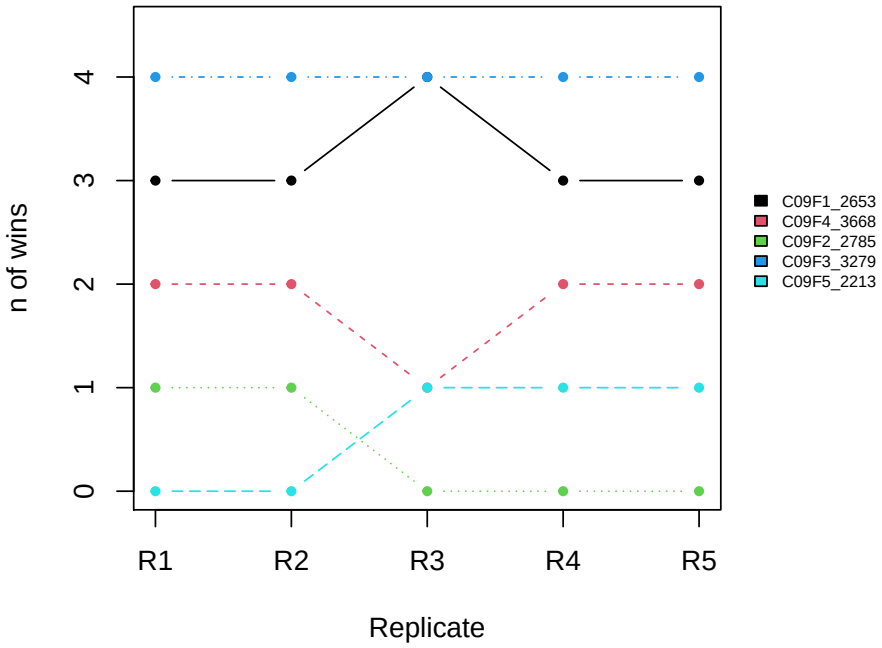
Absolute wins of Cage6 (6M) in FMT per replicate



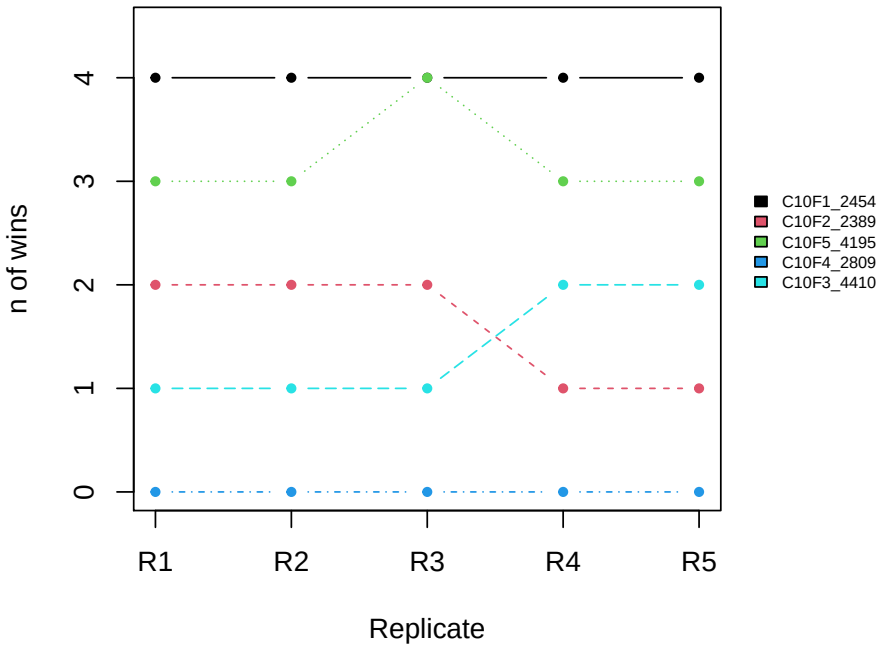

Absolute wins of Cage7 (7M) in FMT per replicate
n of wins
0
1
2
3
4
R1
R2
R3
R4
R5
Replicate
C07M1_2688
C07M2_2730
C07M3_4182
C07M4_3127
C07M5_4178


Absolute wins of Cage8 (8M) in FMT per replicate
n of wins
0
1
2
3
4
R1
R2
R3
R4
R5
Replicate
C08M1_3988
C08M2_3294
C08M4_3065
C08M3_3677
C08M5_3735

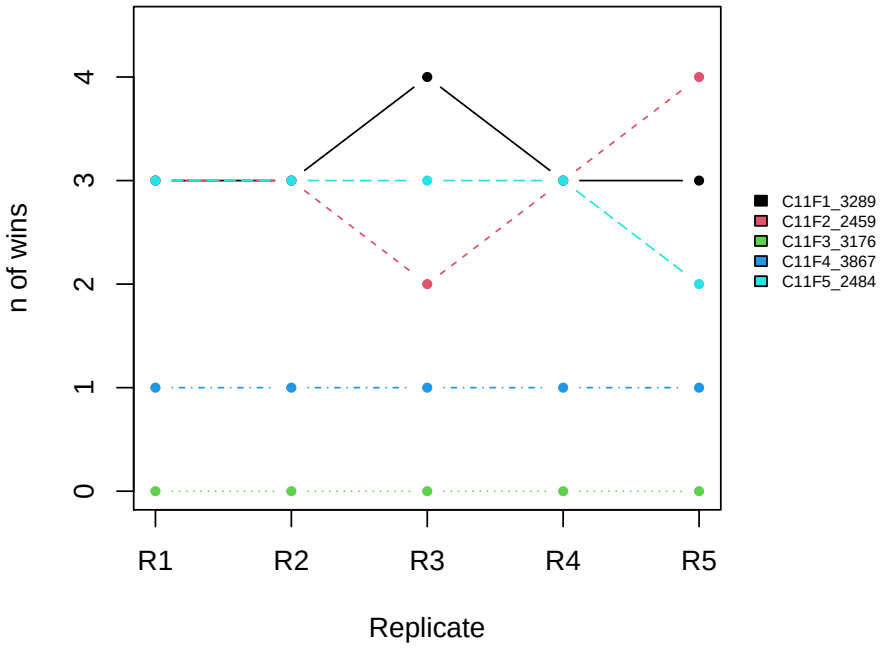
Absolute wins of Cage9 (9F) in FMT per replicate



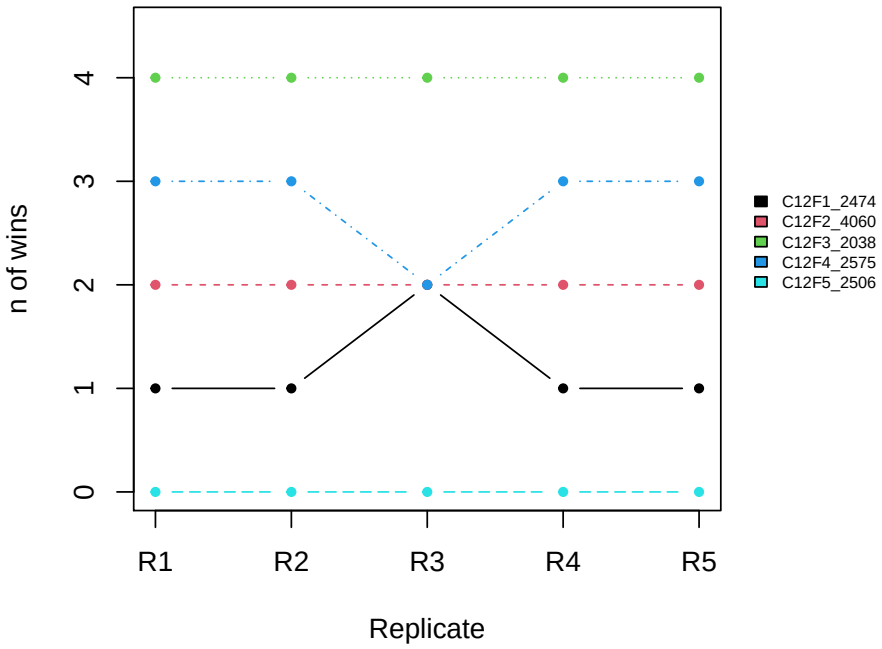
Absolute wins of Cage10 (10F) in FMT per replicate



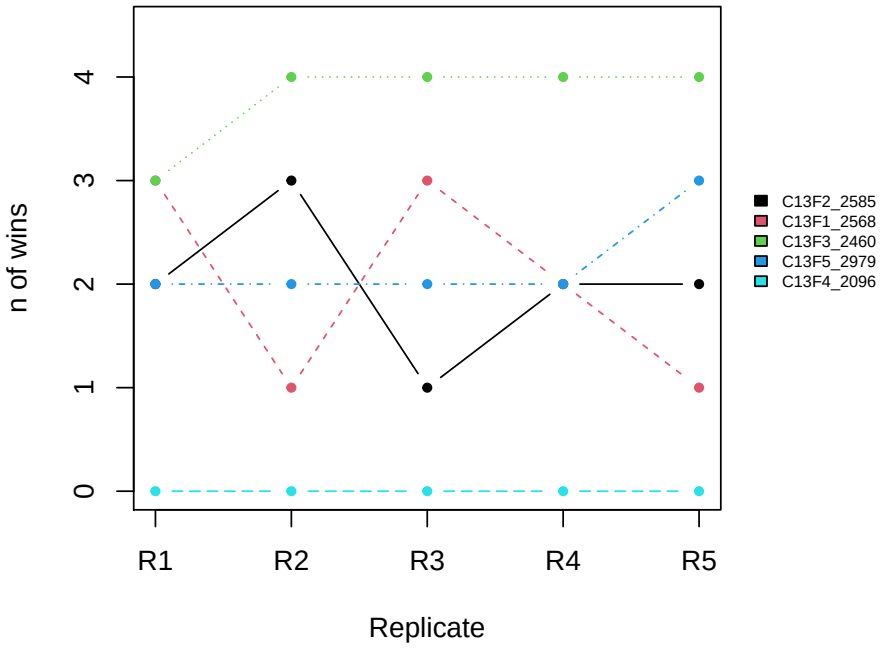
Absolute wins of Cage11 (11F) in FMT per replicate



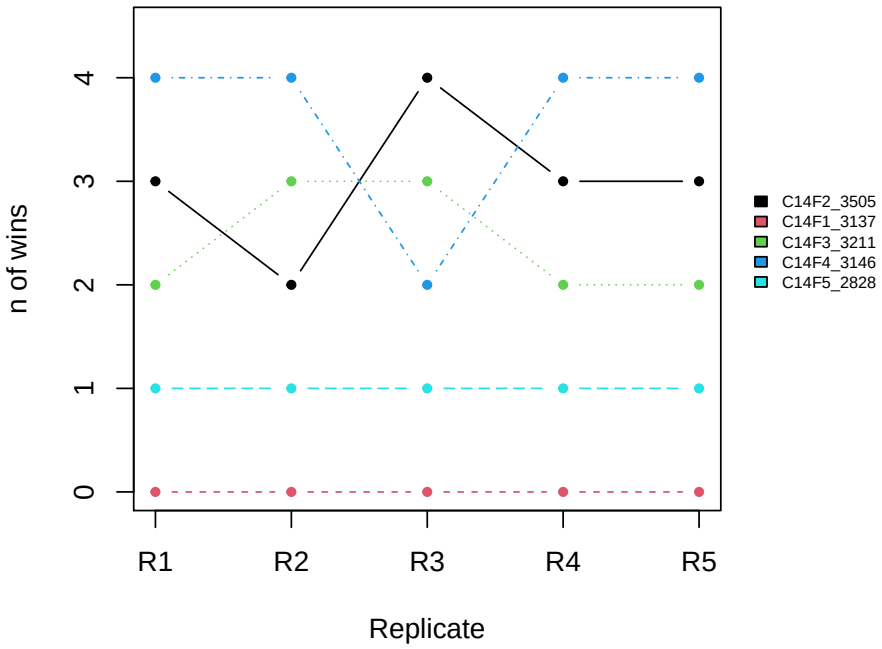
Absolute wins of Cage12 (12F) in FMT per replicate

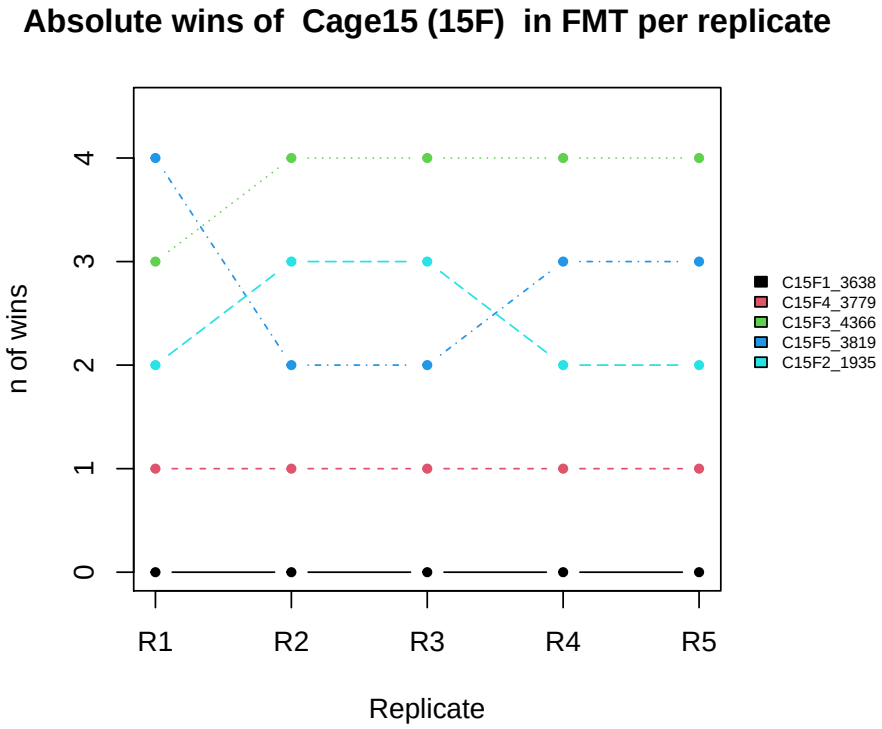


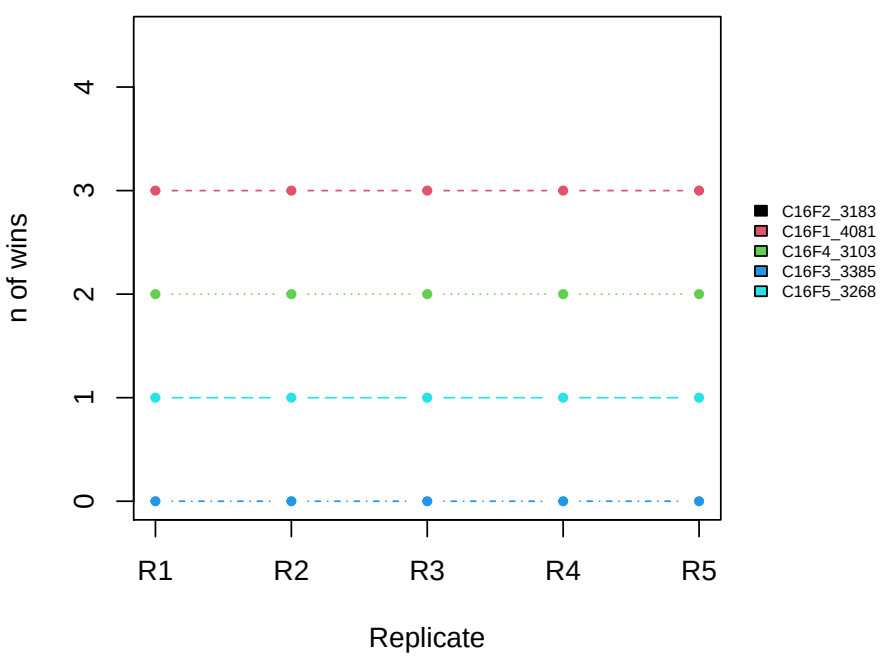
Absolute wins of Cage13 (13F) in FMT per replicate



Absolute wins of Cage14 (14F) in FMT per replicate





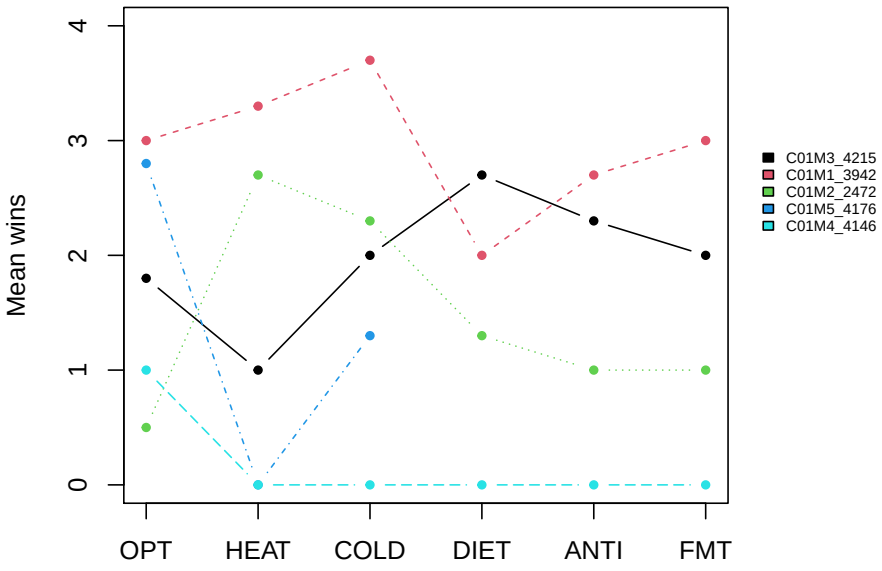


ANSWER TO THE FIRST QUESTION: The dominance hierarchy (the replicates) changes within a treatment per cage. Hence, no the respective cage dominance hierarchies are not stable. They change within tube test replicates for each treatment.

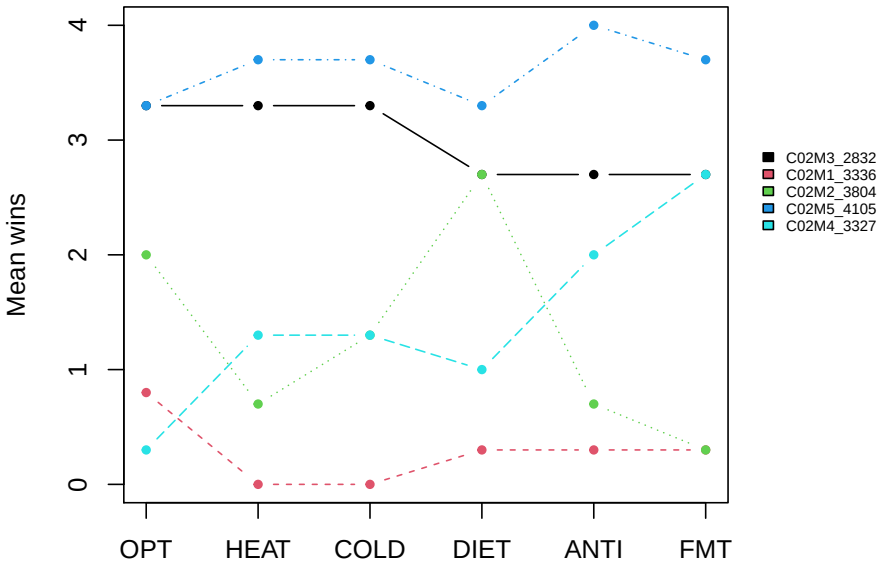
SECOND QUESTION: 2. Is there an overall or stable hierarchy (despite treatments)? For example one mouse continuously winning in the experiment.

I try to answer this by visualizing the average number of wins per mouse per treatment as dot plots.

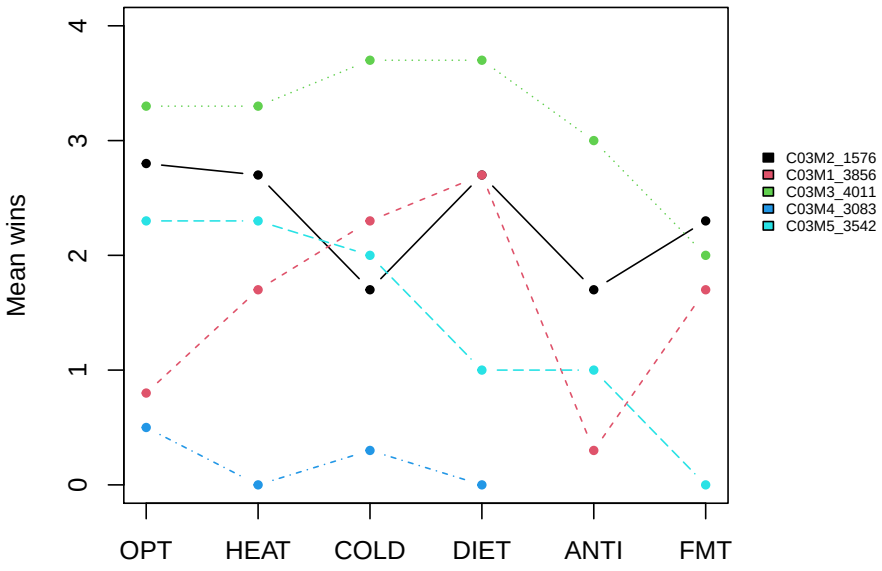
Average wins of each mouse of Cage1 (1M) per treatment



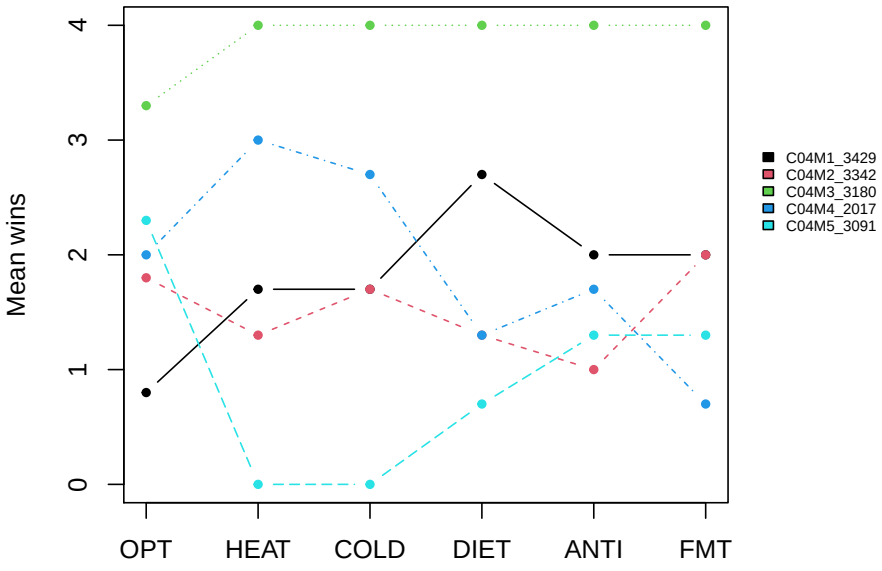
Average wins of each mouse of Cage2 (2M) per treatment



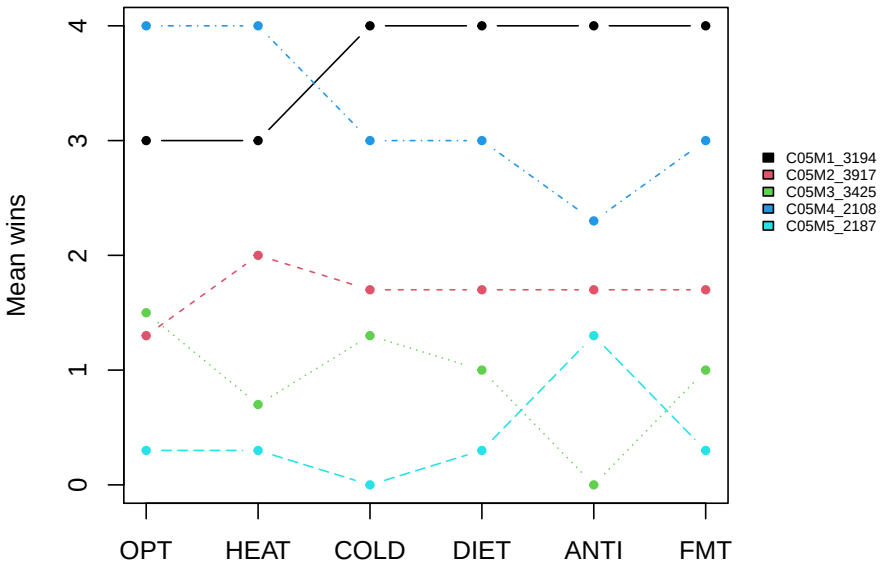
Average wins of each mouse of Cage3 (3M) per treatment



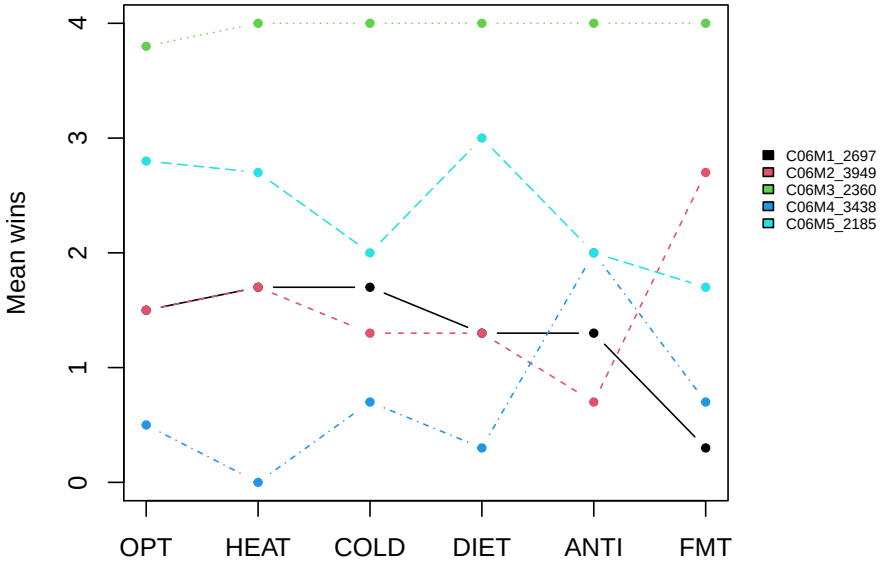
Average wins of each mouse of Cage4 (4M) per treatment



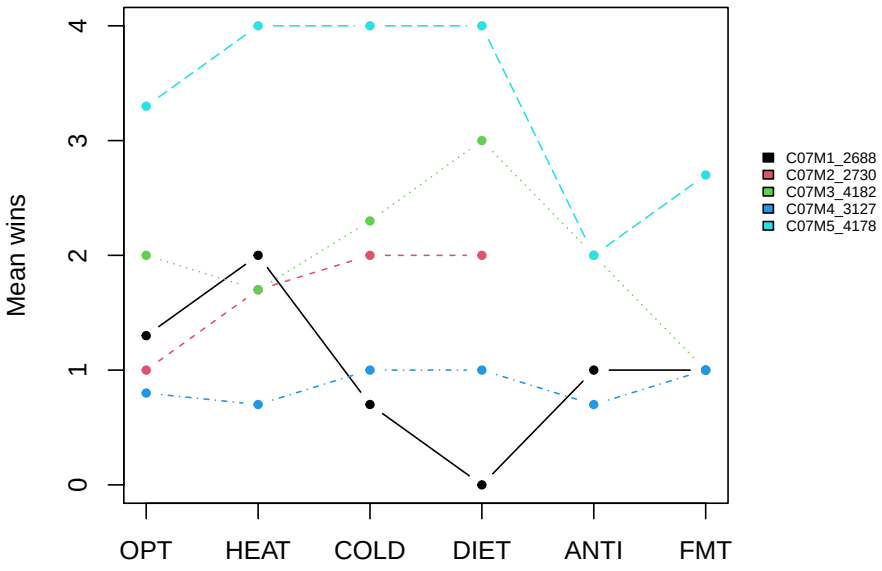
Average wins of each mouse of Cage5 (5M) per treatment



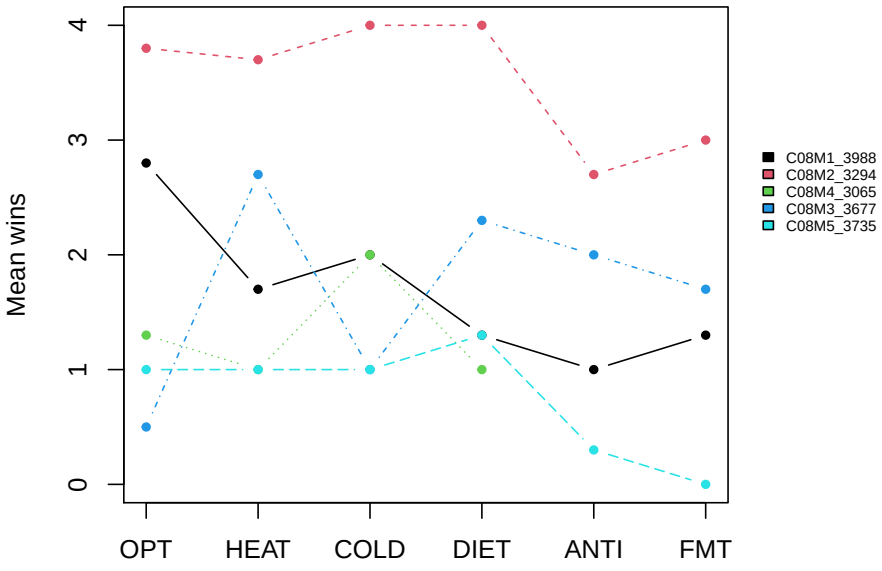
Average wins of each mouse of Cage6 (6M) per treatment



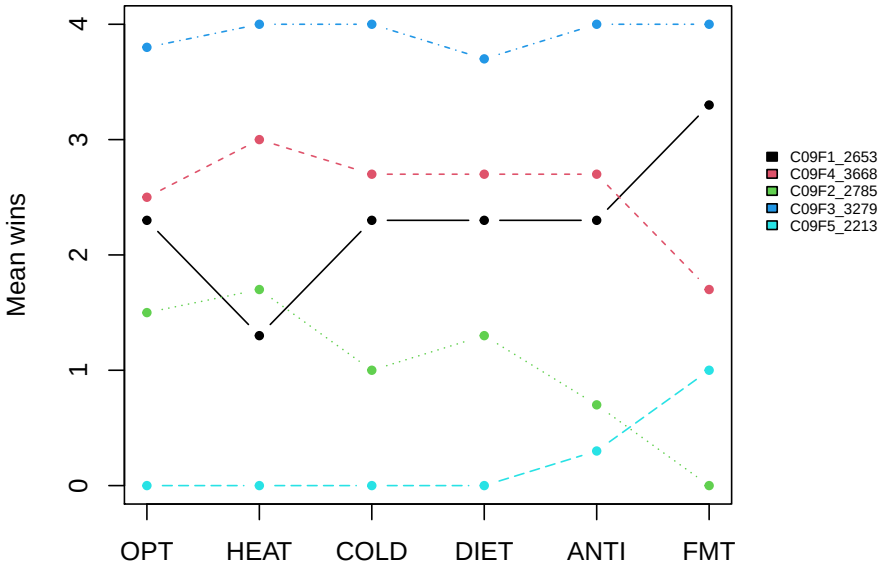
Average wins of each mouse of Cage7 (7M) per treatment



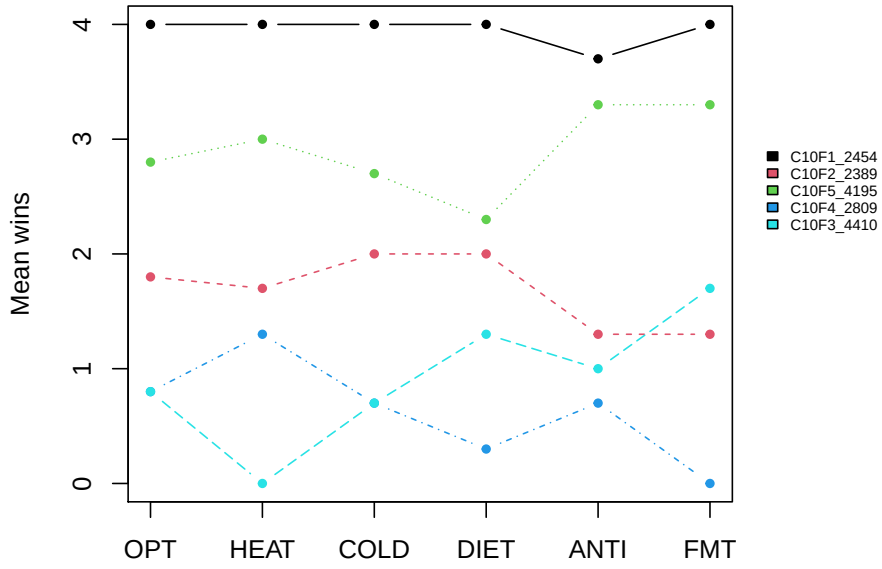
Average wins of each mouse of Cage8 (8M) per treatment



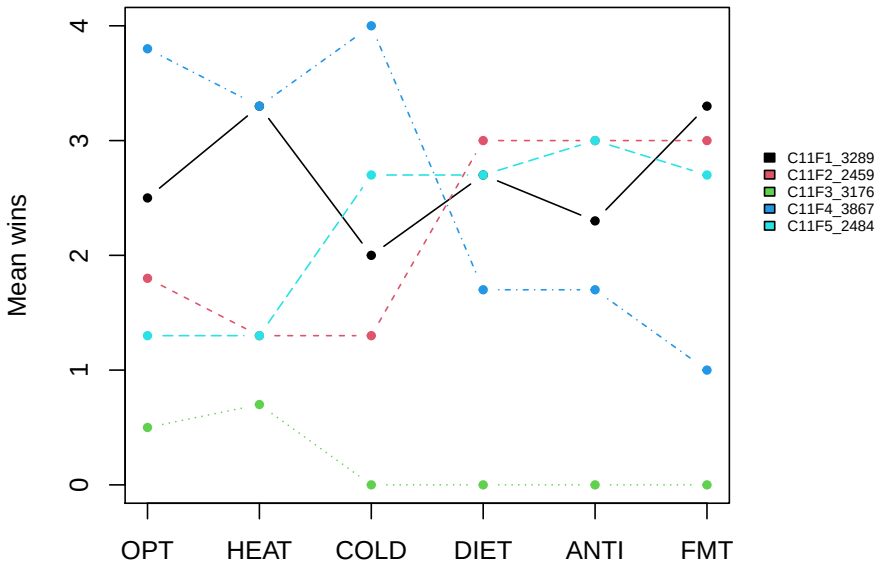
Average wins of each mouse of Cage9 (9F) per treatment



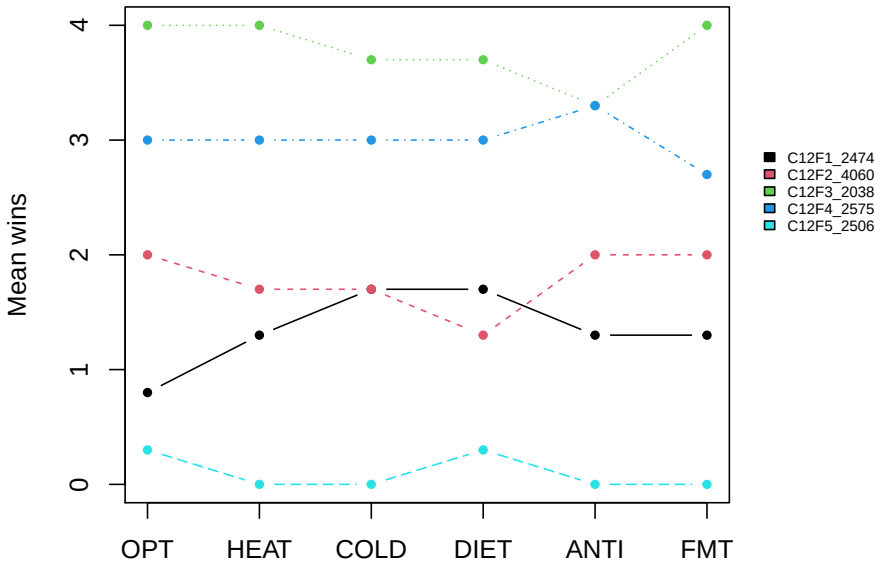
Average wins of each mouse of Cage10 (10F) per treatment



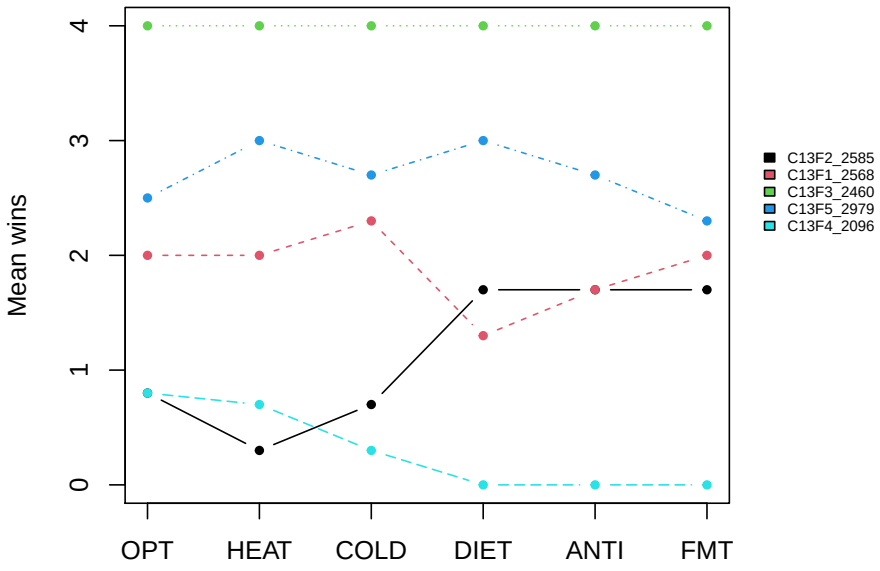
Average wins of each mouse of Cage11 (11F) per treatment



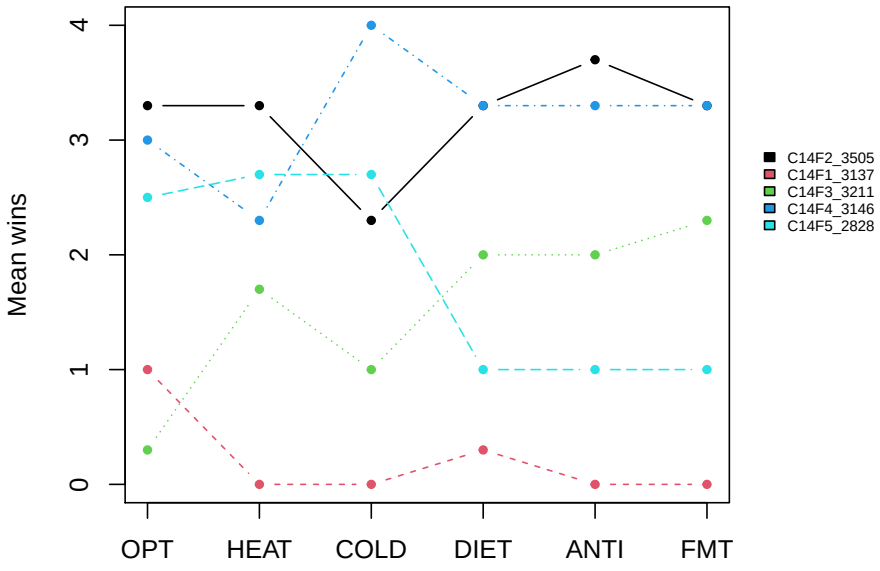
Average wins of each mouse of Cage12 (12F) per treatment

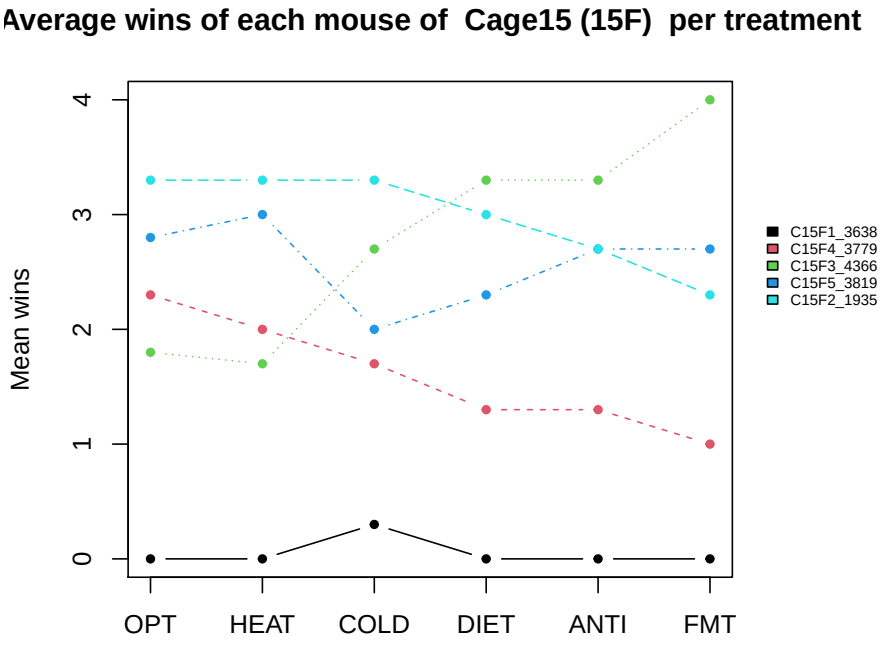


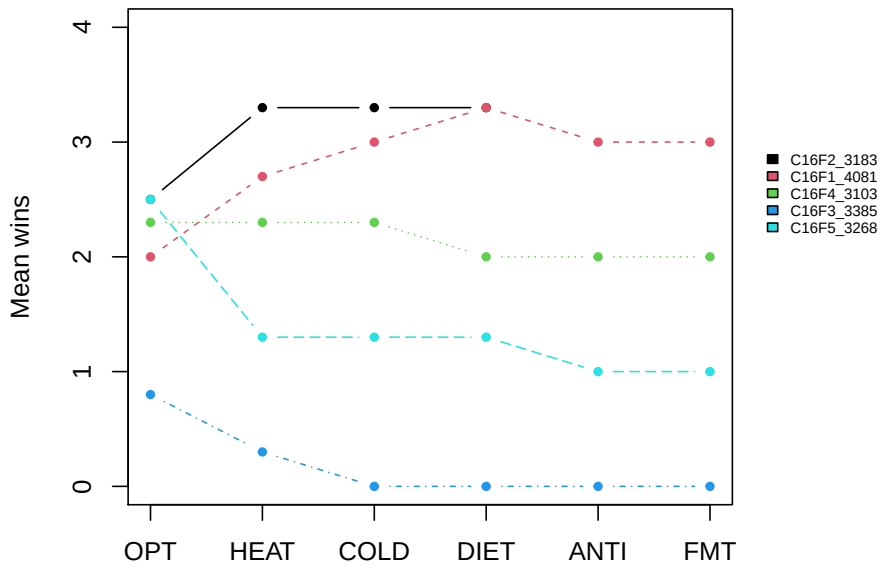
Average wins of each mouse of Cage13 (13F) per treatment



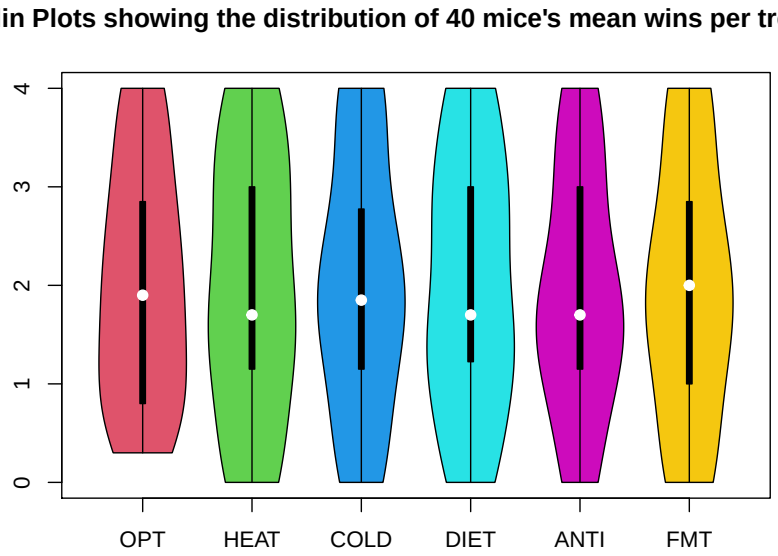
Average wins of each mouse of Cage14 (14F) per treatment



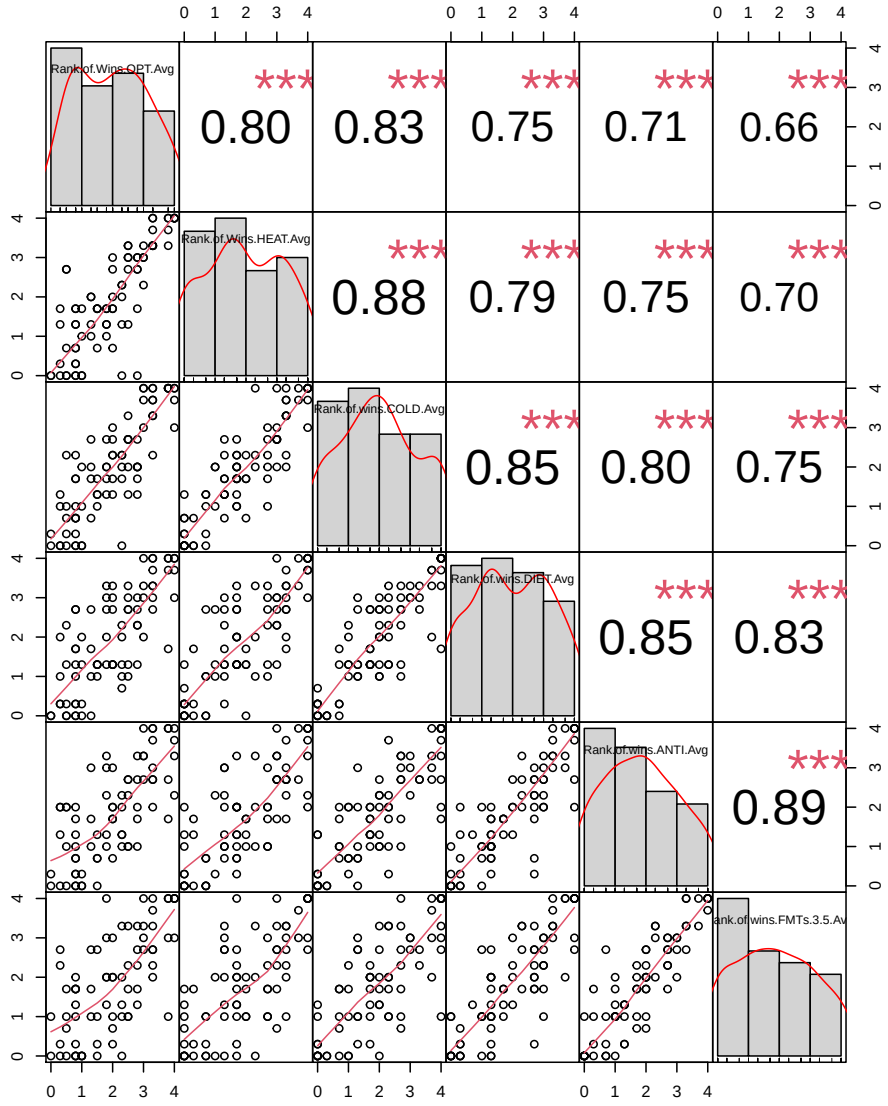




I also try to answer this by visualizing the average number of wins per mouse per treatment as violin plots. If the violin plot look identical, then the dominance hierachy is stable.



Moreover, I tried to answer the question by looking at correlations. I use the mean number of wins per mouse, and it should be individual specific. But please have a look at the code Inaki and let me know.



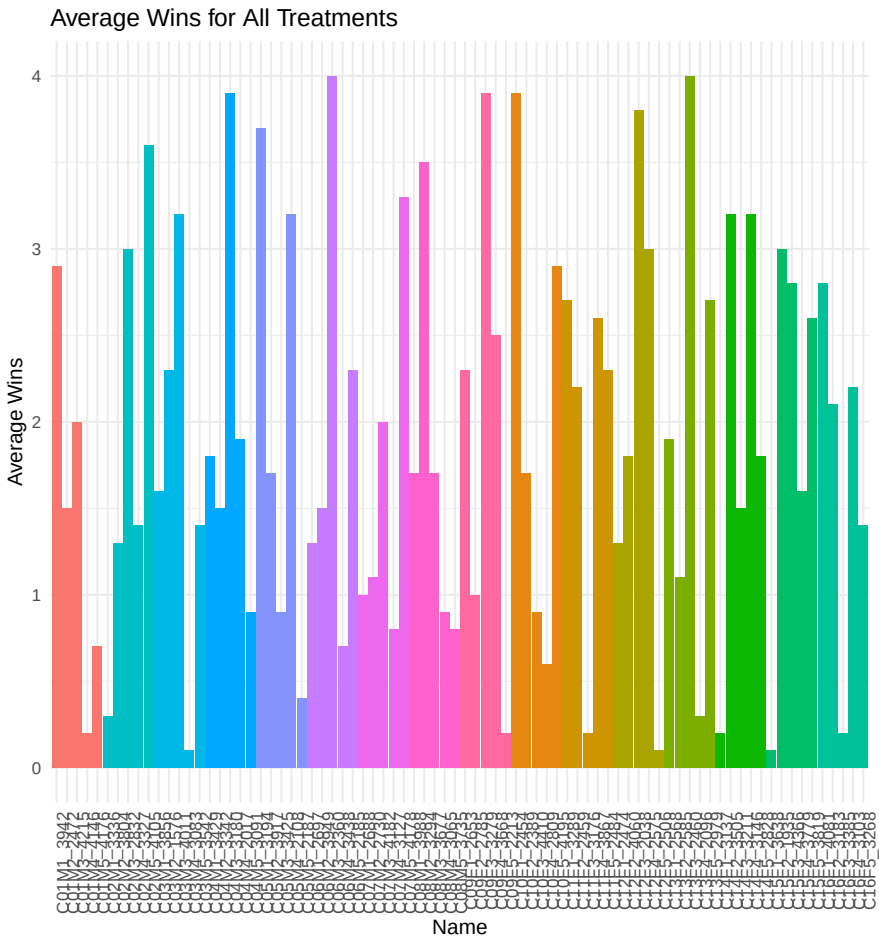
ANSWER TO THE SECOND QUESTION: There is not a stable hierarchy for each cage across treatments. The hierarchy changes for all cages. However, e.g. cage 13 has one mouse dominating from the beginning to the end - on average winning 4 times per treatment.

Nevertheless, it appears that there's an overall hierarchy for each cage across treatments. Thus, that time matters. The correlations on mean wins per mouse show that the treatments close to each other correlate. I was recommended by Antton and Ostaizka oto split the data into e.g. sex and cages, and then do tests, correlation plots etc. Do you agree? Either way, I think it would be smart to do. So I'll work on it this week.

THIRD QUESTION: 3. Is there a difference in dominance between the sex?

ANSWER TO THE THIRD QUESTION: I feel like I've answered this question in the statistical tests using glm() that shows "NA" at sex, which means no significance. But perhaps you Inaki can suggest something?

FOURTH QUESTION: 4. What does the accumulative number of wins look like per mouse within each cage?



ANSWER TO THE FOURTH QUESTION:

We can see that one mouse per cage appears to dominate across the experiment despite treatments according to accumulative number of wins.

I look forward to hear what you think of it all Inaki. Thanks a lot for helping me.