

MoTrPAC Animal data: analysis of the phenotypic data

```
# Set the working directory to the folder with the data
setwd("/Users/David/Desktop/MoTrPAC/march_2019/pheno_data/PASS1A.6M-RLS 0,01/3-Data Sets/")
all_csvs = list.files(".") # get all files in dir
# read all files
csv_data = list()
for(fname in all_csvs){
  csv_data[[fname]] = read.csv(fname,stringsAsFactors = F)
}# supply(csv_data,dim) # check the dimensions of the different datasets
```

Acute tests

```
# Get the acute test data
ac_test_data = csv_data[[which(grepl("Acute.Test",names(csv_data)))]]
dim(ac_test_data)
```

```
## [1] 216 27
```

```
# check the time differences between start and end
test_times = as.difftime(ac_test_data$t_complete) - as.difftime(ac_test_data$t_start)
# table of the values: all except for on are 0.5 hours
table(test_times)
```

```
## test_times
## 0.466666666666667      0.5
##                1      215
```

```
# Get the comment of the sample that is not 0.5h
ac_test_data[test_times!=0.5,"comments"]
```

```
## [1] "Treadmill stopped 28:49 (mm:ss) into the acute bout due to problems with the other rat on the s
```

Nest, we analyze the distances. We hypothesized that these are a function of the shocks given or “errors” in the process.

```
# convert the shock lengths to numbers (seconds)
parse_shocktime<-function(x){
  arr = strsplit(x,split=".")[[1]]
  if(length(arr)<2){return(NA)}
  return(as.numeric(arr[1])*60+as.numeric(arr[2]))
}
tmp_x = ac_test_data$howlongshock
tmp_x = sapply(tmp_x, parse_shocktime)
ac_test_data$howlongshock = tmp_x
rm(tmp_x)

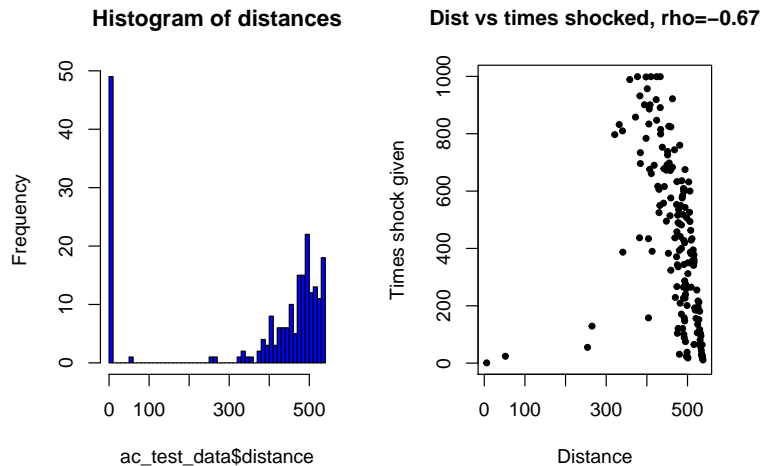
par(mfrow=c(1,2))
# histogram of distances
hist(ac_test_data$distance,col="blue",breaks=50,main = "Histogram of distances")

# Correlation between distance and number of shocks
# Get the indices of the samples with shock information -
# these the animals that did the acute test
```

```

timesshock_inds = !is.na(ac_test_data$timesshock)
# create a new dataframe with the selected animals
trained_animals_data = ac_test_data[timesshock_inds,]
sp_corr = cor(trained_animals_data$distance,
              trained_animals_data$timesshock,method="spearman")
plot(trained_animals_data$distance,trained_animals_data$timesshock,
     main=paste("Dist vs times shocked, rho=",format(sp_corr,digits = 2),sep=""),
     pch=20,ylab="Times shock given",xlab="Distance",cex.main=1.1)

```



```

# A "smarter" analysis: regression of the distance using shock info
dist_lm = lm(distance~timesshock+howlongshock+weight+days_start,
              data=trained_animals_data)
# Summary of the model, points to take: high R^2, significance of
# the features
summary(dist_lm)

```

```

##
## Call:
## lm(formula = distance ~ timesshock + howlongshock + weight +
##     days_start, data = trained_animals_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -412.69   -1.46    6.48   11.87   70.92
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  548.46424   14.11730   38.851  < 2e-16 ***
## timesshock     0.02645    0.01301    2.033  0.0437 *
## howlongshock  -0.29488    0.01983  -14.871  < 2e-16 ***
## weight        -0.20455    0.04091   -5.000  1.47e-06 ***
## days_start     0.21137    0.15813    1.337  0.1832
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 41.24 on 163 degrees of freedom
## Multiple R-squared:  0.6793, Adjusted R-squared:  0.6714
## F-statistic: 86.32 on 4 and 163 DF, p-value: < 2.2e-16

```

```

# We have some clear outliers:
library(MASS)
par(mfrow=c(1,2))
plot(studres(dist_lm),main="studentized residuals (lm)",ylab="residual")
# Select the top outliers and look at their comments
outliers = abs(studres(dist_lm)) > 2
# how many outliers have we selected?
sum(outliers)

```

```
## [1] 4
```

```

# their comments:
trained_animals_data[outliers,"comments"]

```

```

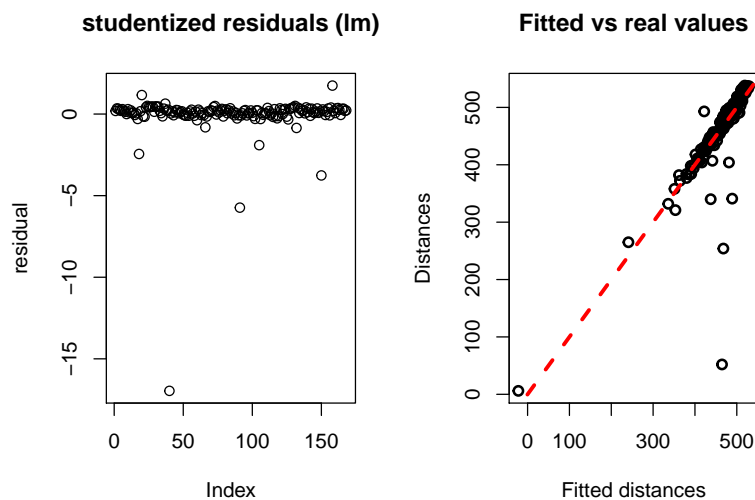
## [1] ""
## [2] "Halfway through the acute test, the shocker grid malfunctioned and would not shut off even when
## [3] "The shocker grid started to malfunction at time point 18:25 of the run. The distance ran, time o
## [4] "Shocker grid malfunctioned. Distance ran, time spent on shocker grid, and number of times rat w

```

```

# Plot the fitted values of the linear regression vs.
# the true distances
plot(dist_lm$fitted.values,trained_animals_data$distance,lwd=2,
     main="Fitted vs real values",ylab="Distances",xlab="Fitted distances")
abline(0,1,col="red",lty=2,lwd=3)

```



Site comparison

```

# Load additional information about the animals
registr_data = csv_data[[which(grepl("Regist",names(csv_data)))]]
rownames(registr_data) = as.character(registr_data$participantGUID)
# make the rownames in the test data comparable
rownames(trained_animals_data) = trained_animals_data$participantGUID
# add sex to the trained animal data data frame
sex_key = c("Female","Male")
trained_animals_data$sex = sex_key[registr_data[rownames(trained_animals_data),"sex"]]

# Map site Ids to their names
site_names = c("910"="Joslin","930"="Florida")
trained_animals_data$site = site_names[as.character(trained_animals_data$siteID)]

```

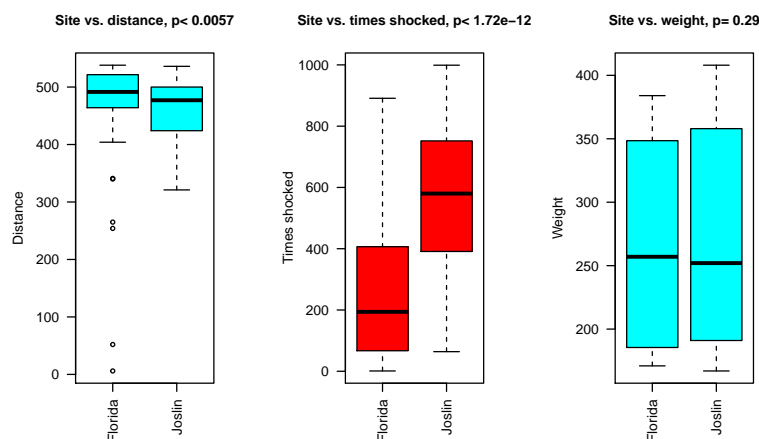
```
# Sanity check: the numbers should be the same for both sites
table(ac_test_data$siteID)
```

```
##
## 910 930
## 108 108
```

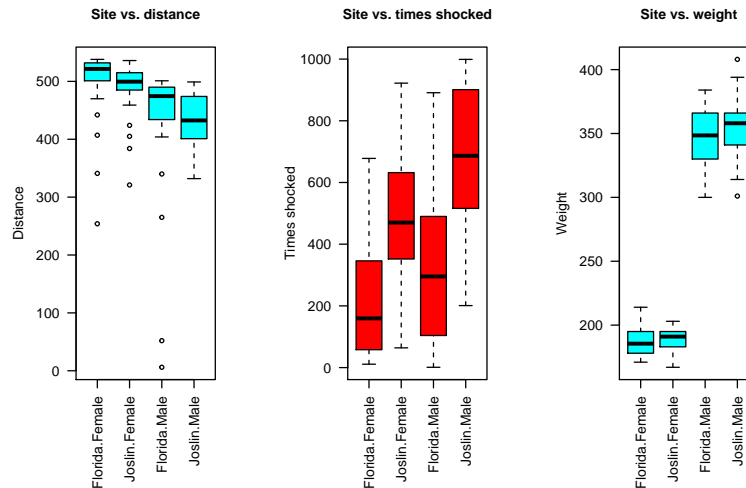
```
table(trained_animals_data$site,trained_animals_data$sex)
```

```
##
##           Female Male
## Florida         42  42
## Joslin          42  42
```

```
run_wilcox<-function(x1,x2){
  return(wilcox.test(x1[x2=="x2[1]],x1[x2!="x2[1]]))$p.value)
}
# Compare the distances, shocks, and weight
par(mfrow=c(1,3),mar=c(10,4,4,4))
# Site only
p_dist = run_wilcox(trained_animals_data$distance,trained_animals_data$site)
boxplot(distance~site,data=trained_animals_data,col="cyan",ylab="Distance",
  main=paste("Site vs. distance, p<",format(p_dist,digits = 2)),
  cex.main=1,las=2)
p_timesshock = run_wilcox(trained_animals_data$timesshock,trained_animals_data$site)
boxplot(timesshock~site,data=trained_animals_data,col="red",ylab="Times shocked",
  main=paste("Site vs. times shocked, p<",format(p_timesshock,digits = 3)),
  cex.main=1,las=2)
p_w = run_wilcox(trained_animals_data$weight,trained_animals_data$site)
boxplot(weight~site,data=trained_animals_data,col="cyan",ylab="Weight",
  main=paste("Site vs. weight, p=",format(p_w,digits = 2)),
  cex.main=1,las=2)
```



```
# Site and sex
par(mfrow=c(1,3),mar=c(10,4,4,4))
boxplot(distance~site+sex,data=trained_animals_data,col="cyan",ylab="Distance",
  main="Site vs. distance",cex.main=1,las=2)
boxplot(timesshock~site+sex,data=trained_animals_data,col="red",ylab="Times shocked",
  main="Site vs. times shocked",cex.main=1,las=2)
boxplot(weight~site+sex,data=trained_animals_data,col="cyan",ylab="Weight",
  main="Site vs. weight",cex.main=1,las=2)
```



Regress time shocked and distance vs. site and sex

```
summary(lm(timesshock~site+sex,data=trained_animals_data))
```

```
##
## Call:
## lm(formula = timesshock ~ site + sex, data = trained_animals_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -453.06 -167.17  -11.96   151.83   543.99
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    197.92     31.02   6.380 1.72e-09 ***
## siteJoslin     307.05     35.82   8.572 7.02e-15 ***
## sexMale        149.10     35.82   4.162 5.06e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 232.1 on 165 degrees of freedom
## Multiple R-squared:  0.355, Adjusted R-squared:  0.3471
## F-statistic: 45.4 on 2 and 165 DF, p-value: < 2.2e-16
```

```
summary(lm(distance~site+sex,data=trained_animals_data))
```

```
##
## Call:
## lm(formula = distance ~ site + sex, data = trained_animals_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -435.20  -10.88    15.44    33.70    67.24
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    501.780     8.745  57.381 < 2e-16 ***
## siteJoslin     -9.440     10.097  -0.935   0.351
## sexMale       -60.583     10.097  -6.000 1.21e-08 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## Residual standard error: 65.44 on 165 degrees of freedom  
## Multiple R-squared:  0.1827, Adjusted R-squared:  0.1727  
## F-statistic: 18.44 on 2 and 165 DF,  p-value: 5.937e-08
```