

Biodistribution

Zach McKay

2022-09-22

Import and view data

We can see that the data structure has caused the column names to not be imported correctly, and that the observation point is not recorded for each observation but rather at the top of each set of observations.

```
## New names:
## Rows: 64 Columns: 19
## -- Column specification
## ----- Delimiter: "," chr
## (18): May '22, ...2, ...3, ...4, ...5, ...6, ...7, ...8, ...9, ...10, ..... dbl
## (1): ...19
## i Use 'spec()' to retrieve the full column specification for this data. i
## Specify the column types or set 'show_col_types = FALSE' to quiet this message.
## * ' -> '...2'
## * ' -> '...3'
## * ' -> '...4'
## * ' -> '...5'
## * ' -> '...6'
## * ' -> '...7'
## * ' -> '...8'
## * ' -> '...9'
## * ' -> '...10'
## * ' -> '...11'
## * ' -> '...12'
## * ' -> '...13'
## * ' -> '...14'
## * ' -> '...15'
## * ' -> '...16'
## * ' -> '...17'
## * ' -> '...18'
## * ' -> '...19'
```

```
## # A tibble: 6 x 19
##   'May '22' ...2 ...3 ...4 ...5 ...6 ...7 ...8 ...9 ...10 ...11 ...12
##   <chr>      <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr>
## 1 Mouse no. Ltr   Intrvl Ser pfu Spln~ Spln~ Ipsi~ Ipsi~ Cont~ Cont~ Kidn~ Kidn~
## 2 539      A    24hpi 0      90    2037 170    2941~ 120    33333 180    19
## 3 542      B    24hpi 0      100   333   190    1403~ 160    25000 170    59
## 4 548      C    24hpi 0      110   240   230    2391~ 260    51282 180    130
## 5 546      D    24hpi 0      130   256   180    1388~ 160    25000 200    500
```

```
## 6 552      E      24hpi 0      60      333      180      1111~ 150      2222      180      0
## # ... with 7 more variables: ...13 <chr>, ...14 <chr>, ...15 <chr>,
## #   ...16 <chr>, ...17 <chr>, ...18 <chr>, ...19 <dbl>
## # i Use 'colnames()' to see all variable names
```

Separate data sets into separate objects and clean up data

We will clean up the data by first splitting the data into three separate sets so that each observation point can be cleaned up for later recombination.

```
## # A tibble: 6 x 19
##   'May '22' ...2 ...3 ...4 ...5 ...6 ...7 ...8 ...9 ...10 ...11 ...12
##   <chr>      <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr>
## 1 Mouse no. Ltr   Intrvl Ser pfu Spln~ Spln~ Ipsi~ Ipsi~ Cont~ Cont~ Kidn~ Kidn~
## 2 539      A     24hpi 0      90      2037 170 2941~ 120 33333 180 19
## 3 542      B     24hpi 0      100     333 190 1403~ 160 25000 170 59
## 4 548      C     24hpi 0      110     240 230 2391~ 260 51282 180 130
## 5 546      D     24hpi 0      130     256 180 1388~ 160 25000 200 500
## 6 552      E     24hpi 0      60      333 180 1111~ 150 2222 180 0
## # ... with 7 more variables: ...13 <chr>, ...14 <chr>, ...15 <chr>,
## #   ...16 <chr>, ...17 <chr>, ...18 <chr>, ...19 <dbl>
## # i Use 'colnames()' to see all variable names
```

```
## # A tibble: 6 x 19
##   'May '22' ...2 ...3 ...4 ...5 ...6 ...7 ...8 ...9 ...10 ...11 ...12
##   <chr>      <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr>
## 1 July '22 <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA>
## 2 Mouse no. Ltr   Intrvl Spln (~ plaq~ Spln~ Ipsi~ plaq~ Ipsi~ Cont~ plaq~ Cont~
## 3 1827      A     24      100      0      0     135 9x10~ 2250~ 180 4x10~ 74074
## 4 1828      B     24      110     200 6060 130 5x10~ 1950~ 150 3x10~ 66667
## 5 1830      C     24      100     40 1333 175 6x10~ 1142~ 160 6x10~ 1250~
## 6 1833      D     24      80      ND      ND     180 6x10~ 1111~ 160 7x10~ 1458~
## # ... with 7 more variables: ...13 <chr>, ...14 <chr>, ...15 <chr>,
## #   ...16 <chr>, ...17 <chr>, ...18 <chr>, ...19 <dbl>
## # i Use 'colnames()' to see all variable names
```

```
## # A tibble: 6 x 19
##   'May '22' ...2 ...3 ...4 ...5 ...6 ...7 ...8 ...9 ...10 ...11 ...12
##   <chr>      <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr>
## 1 August '22 ~ <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA>
## 2 Mouse no. Ltr   Intr~ Ipsi~ plaq~ Ipsi~ Cont~ plaq~ Cont~ CLN ~ plaq~ CLN ~
## 3 Ntg      A     120 160 9x10~ 1873 150 4x10~ 889 15 <NA> 0
## 4 Ntg      B     120 180 8x10~ 1481 200 5x10~ 833 15 <NA> 0
## 5 Ntg      C     120 150 4x10~ 889 160 2x10~ 4162 15 <NA> 2200
## 6 Ntg      D     120 140 2.5x~ 595 160 3x10~ 6244 20 <NA> 0
## # ... with 7 more variables: ...13 <chr>, ...14 <chr>, ...15 <chr>,
## #   ...16 <chr>, ...17 <chr>, ...18 <chr>, ...19 <dbl>
## # i Use 'colnames()' to see all variable names
```

Clean up data

We need to rename the columns to useful descriptors for each observation. We will use the mouse, ltr (for letter), and total pfu, weight, and calculate pfu/g based on what is available for each tissue type. We will then delete blank or unnecessary rows (rows that currently contain data type), and an additional column for the observaiton point for later recombinaiton.

```
## # A tibble: 6 x 15
##   mouse ltr   intrvl ser_pfu spleen_mg spleen_~1 ipsi_mg ipsi_~2 contr~3 contr~4
##   <chr> <chr> <chr>   <chr>   <chr>   <chr>   <chr>   <chr>   <chr>   <chr>
## 1 539   A    24hpi    0      90     2037    170    2941176 120    33333
## 2 542   B    24hpi    0     100     333     190    1403508 160    25000
## 3 548   C    24hpi    0     110     240     230    2391304 260    51282
## 4 546   D    24hpi    0     130     256     180    1388888 160    25000
## 5 552   E    24hpi    0      60     333     180    1111111 150     2222
## 6 545   F    48hpi    0     120      0     180    222222 170     3137
## # ... with 5 more variables: kidney_mg <chr>, kidney_pfu_g <chr>, cln_mg <chr>,
## #   cln_pfu_g <chr>, obs_pt <chr>, and abbreviated variable names
## #   1: spleen_pfu_g, 2: ipsi_pfu_g, 3: contra_mg, 4: contra_pfu_g
## # i Use 'colnames()' to see all variable names
```

We will perform similar data cleanup for the July data set.

```
## # A tibble: 6 x 20
##   mouse ltr   intrvl spleen_mg spleen_~1 splee~2 ipsi_mg ipsi_~3 ipsi_~4 contr~5
##   <chr> <chr> <chr>   <chr>   <chr>   <chr>   <chr>   <chr>   <chr>   <chr>
## 1 1827   A     24     100      0      0     135    9x10e4~ 2250000 180
## 2 1828   B     24     110     200    6060    130    5x10e4~ 1950195 150
## 3 1830   C     24     100     40    1333    175    6x10e5~ 114285~ 160
## 4 1833   D     24     80      ND     ND     180    6x10e5~ 111111~ 160
## 5 1835   E     24     90      ND     ND     170    5x10e5~ 9803922 150
## 6 1838   F     48     120     ND     ND     150    3x10e4~ 666667 170
## # ... with 10 more variables: contra_plaques <chr>, contra_pfu_g <chr>,
## #   kidney_mg <chr>, kidney_plaques <chr>, kidney_pfu_g <chr>, cln_mg <chr>,
## #   cln_plaques <chr>, cln_pfu_g <chr>, ...19 <dbl>, obs_pt <chr>, and
## #   abbreviated variable names 1: spleen_plaques, 2: spleen_pfu_g,
## #   3: ipsi_plaques, 4: ipsi_pfu_g, 5: contra_mg
## # i Use 'colnames()' to see all variable names
```

And the same process for the August data set.

```
## # A tibble: 6 x 13
##   mouse ltr   intrvl ipsi_mg ipsi_plaques ipsi_~1 contr~2 contr~3 contr~4 cln_mg
##   <chr> <chr> <chr>   <chr>   <chr>   <chr>   <chr>   <chr>   <chr>
## 1 Ntg   A     120    160    9x10e1    1873    150    4x10e1~ 889    15
## 2 Ntg   B     120    180    8x10e1    1481    200    5x10e1~ 833    15
## 3 Ntg   C     120    150    4x10e1/300uL 889    160    2x10e2~ 4162   15
## 4 Ntg   D     120    140    2.5x10e1/30~ 595    160    3x10e2~ 6244   20
## 5 Ntg   E     120    150    4x10e2/300uL 8888    180    2.5x10~ 463    15
## 6 Tg     A     120    160    6x10e2/300uL 12499    170    1x10e1~ 195    20
## # ... with 3 more variables: cln_plaques <chr>, cln_pfu_g <chr>, obs_pt <chr>,
## #   and abbreviated variable names 1: ipsi_pfu_g, 2: contra_mg,
## #   3: contra_plaques, 4: contra_pfu_g
## # i Use 'colnames()' to see all variable names
```

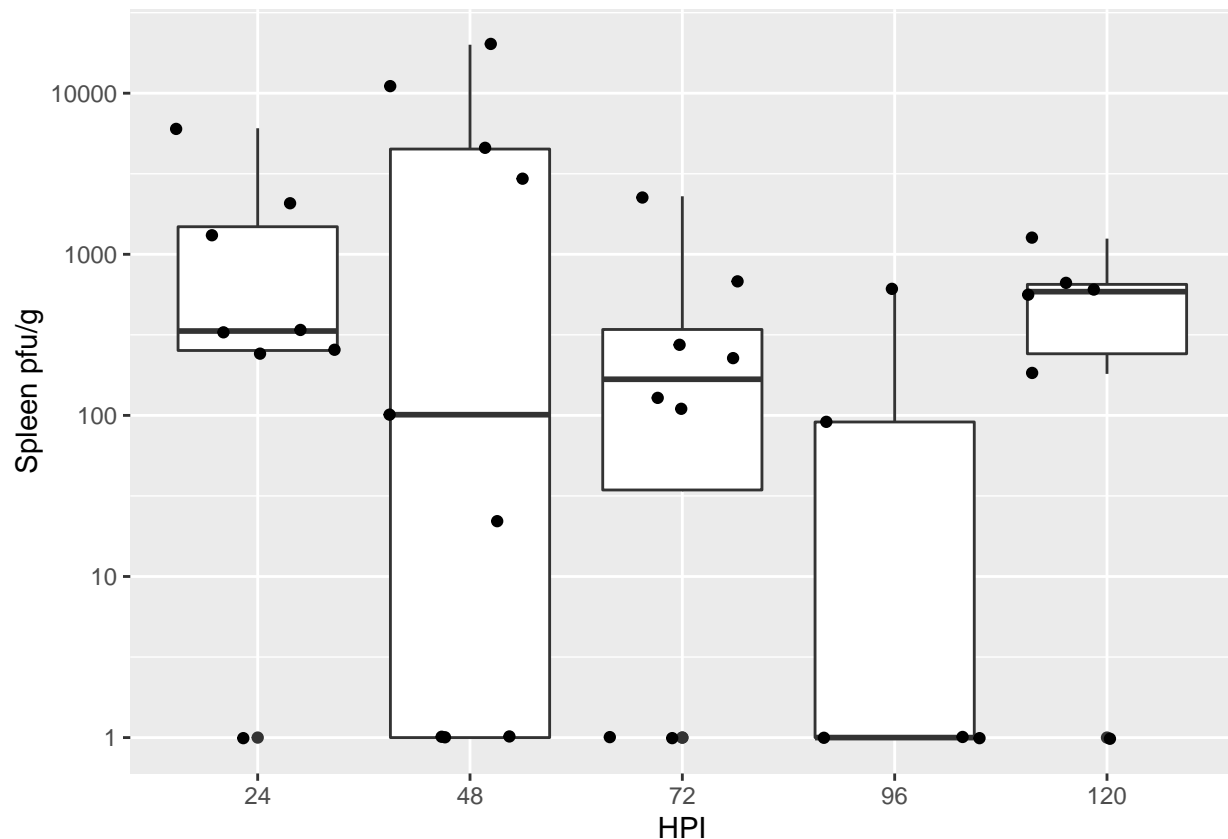
Combine data into new set

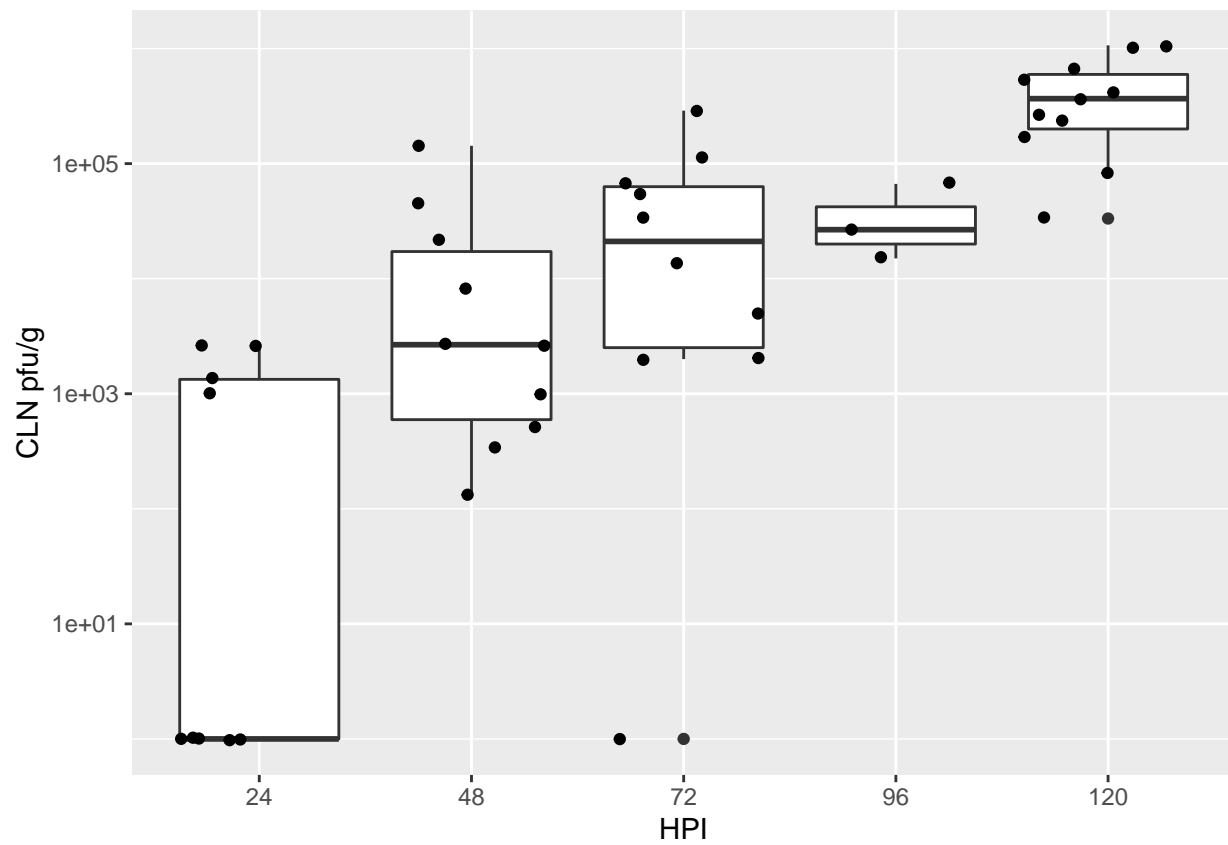
Now we can recombine the cleaned up data sets into a single data set. We also want to convert each data type to numeric where possible and convert some observations to n/a. Also, we will perform our analyses using logarithmic scales, so to correct for 0 values, we will perform our analyses by using $x+1$, so each numeric value will have 1 added to it.

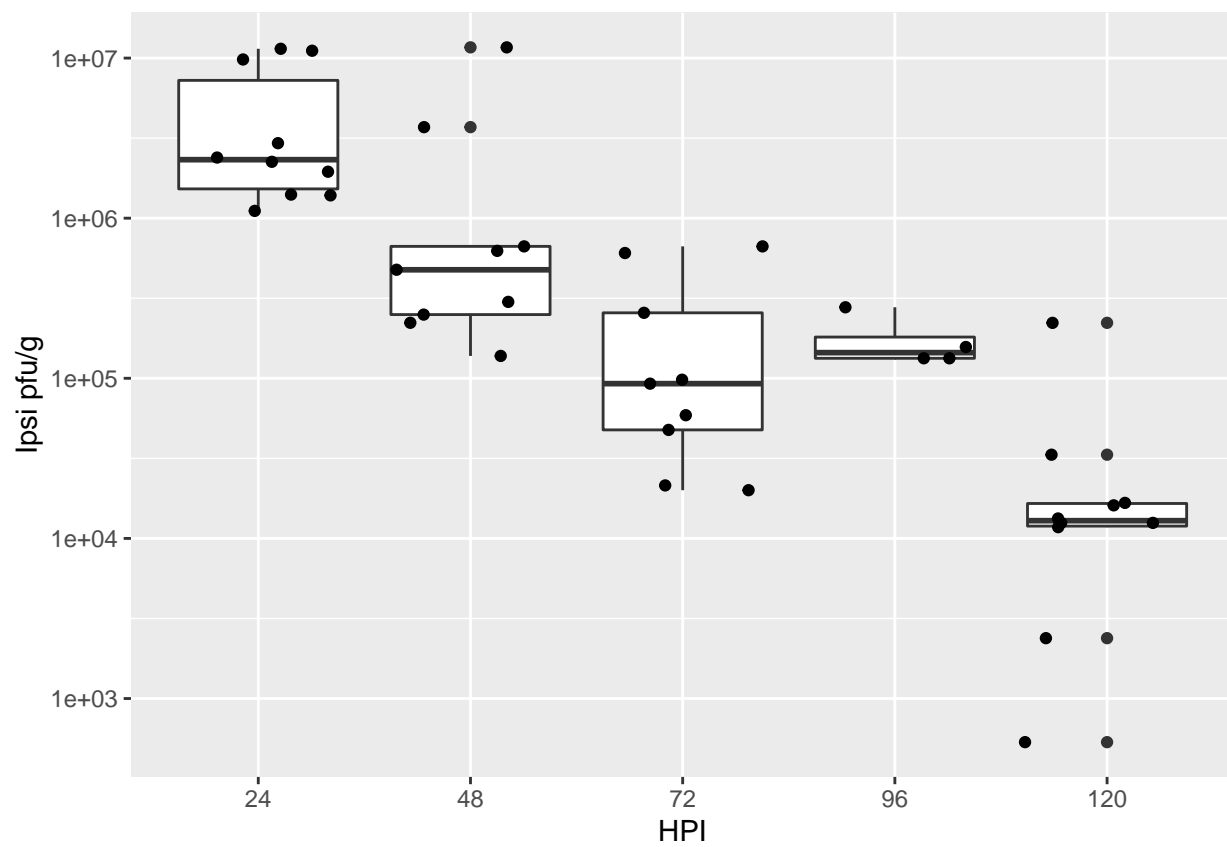
```
## Joining, by = c("mouse", "ltr", "intrvl", "spleen_mg", "spleen_pfu_g",  
## "ipsi_mg", "ipsi_pfu_g", "contra_mg", "contra_pfu_g", "kidney_mg",  
## "kidney_pfu_g", "cln_mg", "cln_pfu_g", "obs_pt")  
## Joining, by = c("mouse", "ltr", "intrvl", "ipsi_mg", "ipsi_pfu_g", "contra_mg",  
## "contra_pfu_g", "cln_mg", "cln_pfu_g", "obs_pt", "ipsi_plaques",  
## "contra_plaques", "cln_plaques")
```

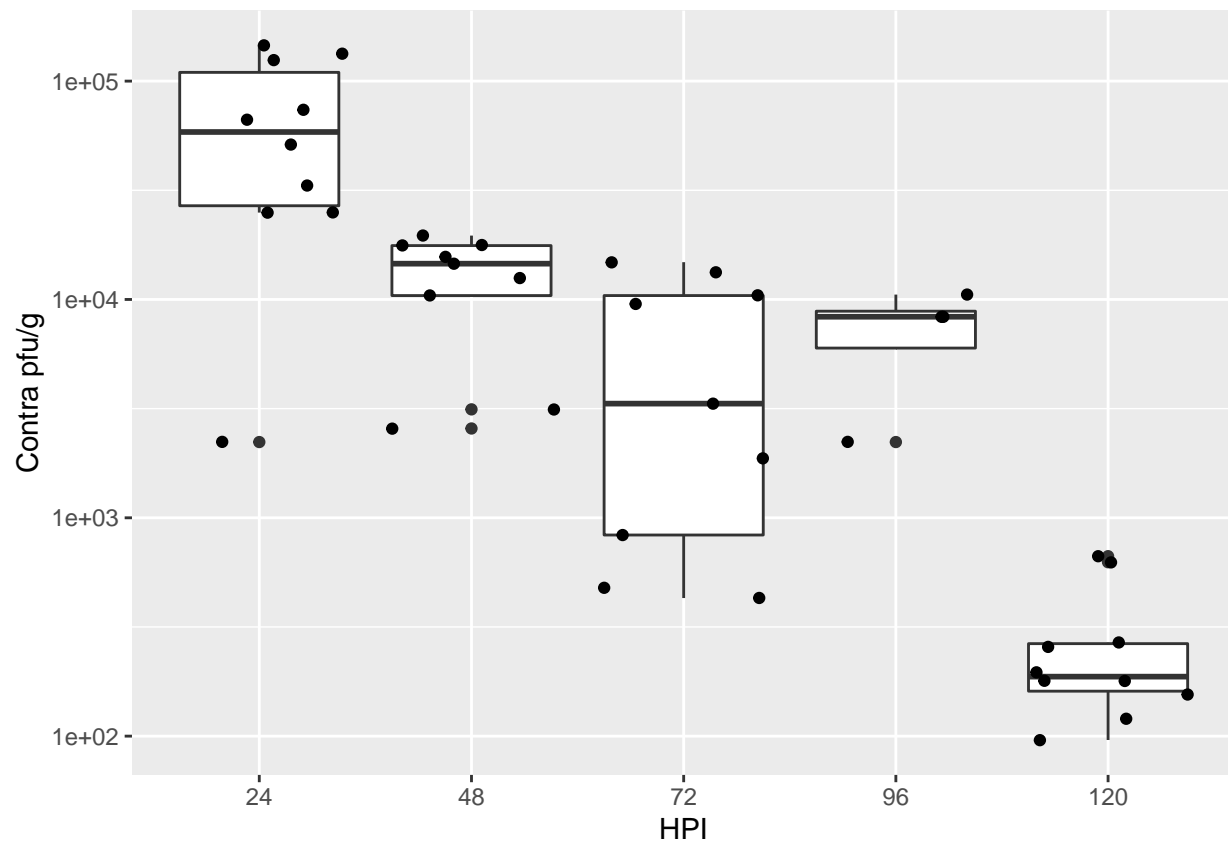
Create and save plots by tissue type

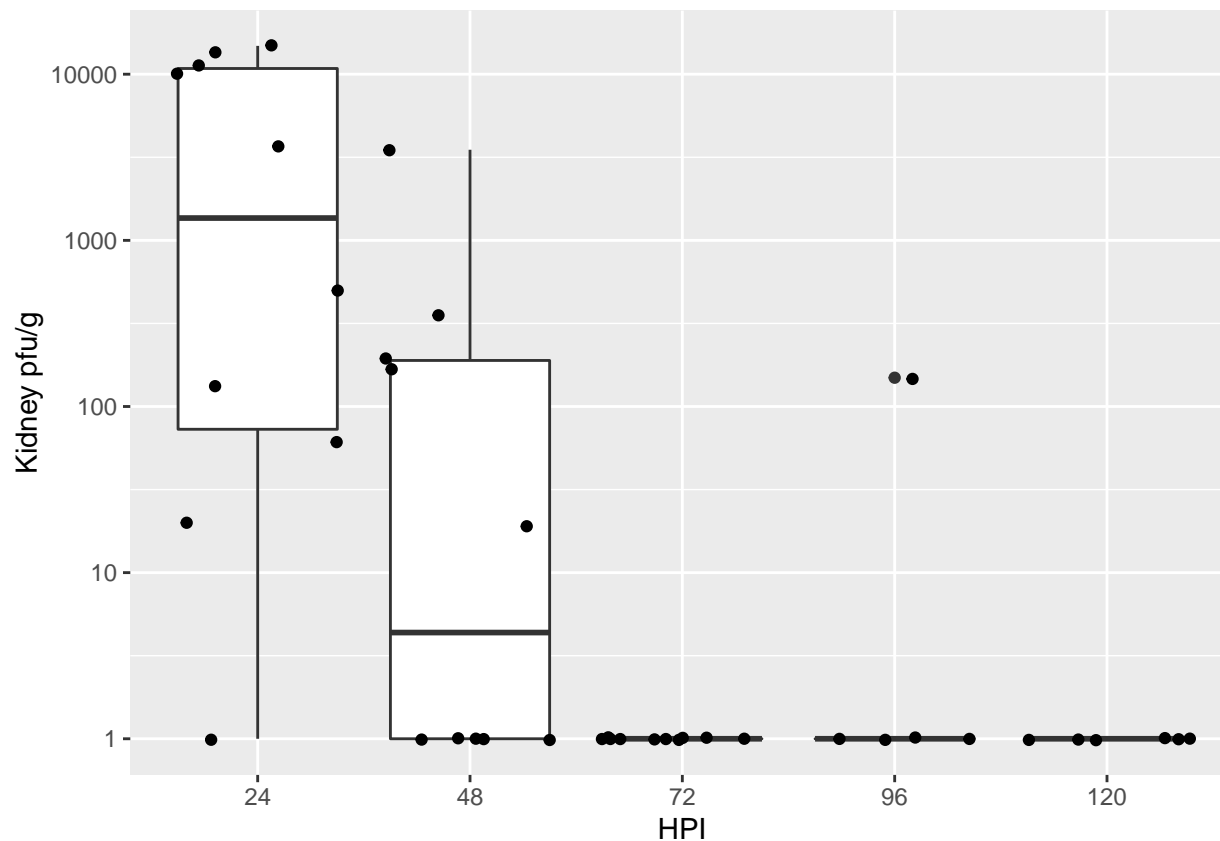
We look at the data pooled across observation points for each graph by tissue type and separate the data by time point. Since the August data set also includes the non-transgenic mice, we will filter those mice out for the final analysis. Each graph will be saved for later use











Combine all data into single graph

We will create a data set from the biodistribution data set that only includes the pfu/g and necessary comparators (intrvl, mouse, ltr, obs_pt).

```
## # A tibble: 6 x 6
##   intrvl mouse ltr   obs_pt tissue   pfu_g
##   <dbl> <chr> <chr> <chr> <chr>   <dbl>
## 1    24 539   A   may_22 Spleen    2038
## 2    24 539   A   may_22 Ipsi  2941177
## 3    24 539   A   may_22 Contra 33334
## 4    24 539   A   may_22 Kidney    20
## 5    24 539   A   may_22 CLN      1
## 6    24 542   B   may_22 Spleen    334
```

Plot all tissue against each other

We can then plot all of the tissues onto a single graph. We will filter out all of the non-transgenic mice into this analysis as done previously.

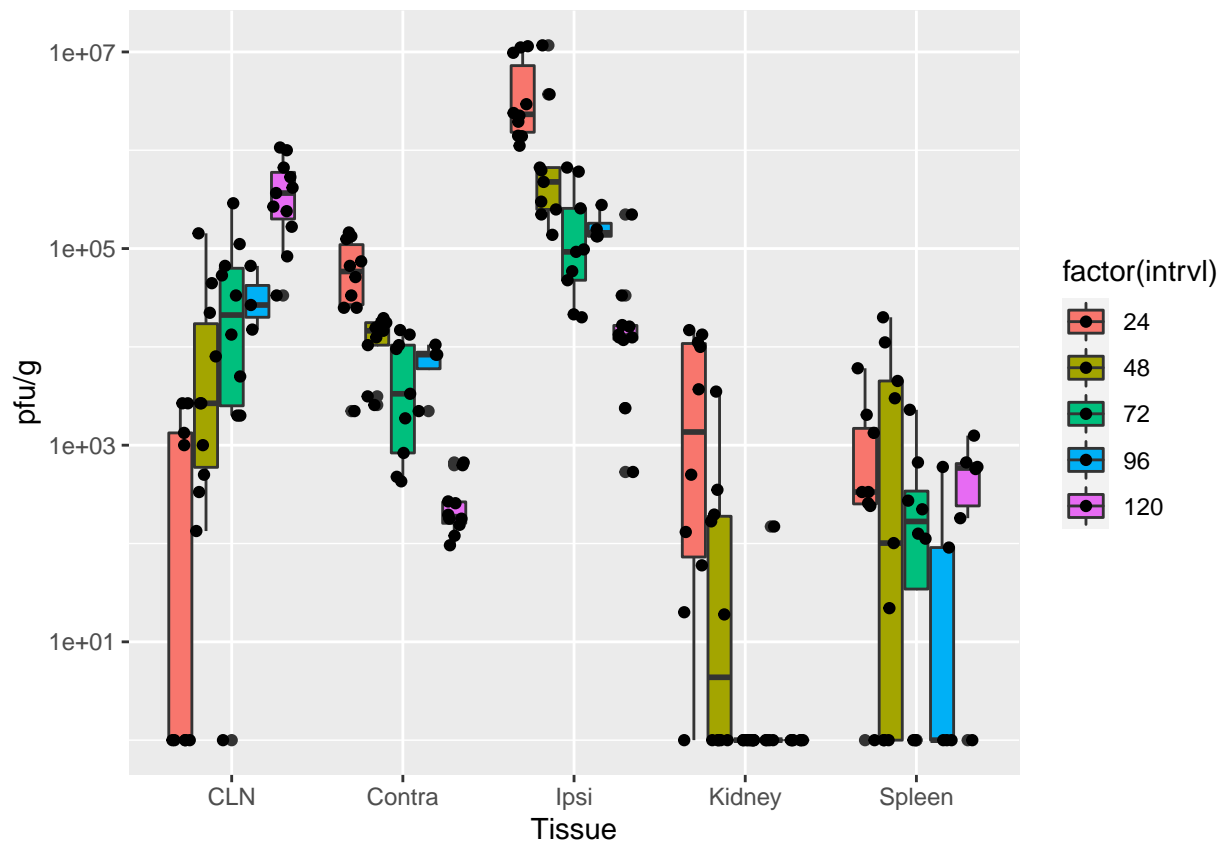


Figure 1 - July only CLN, brain, and kidney

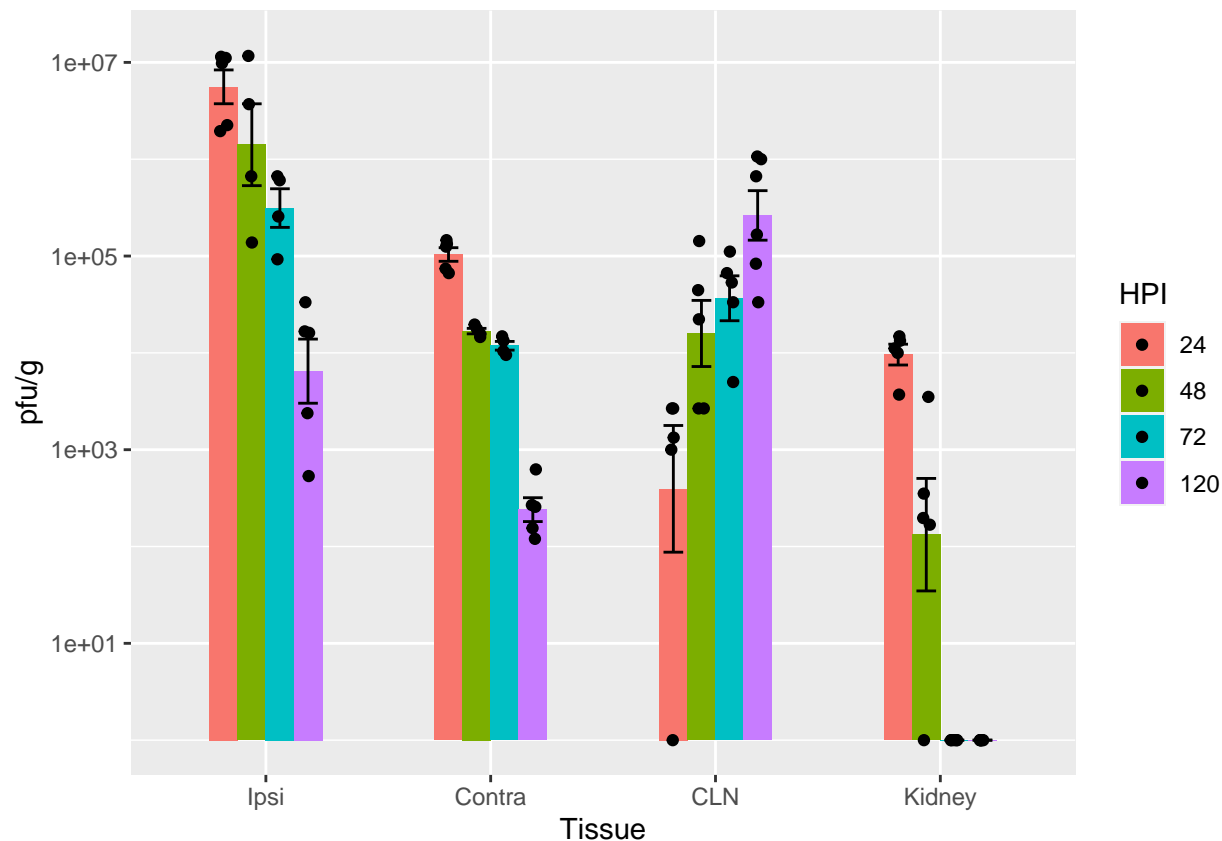
We will look at only July for the timecourse, so we will create a data set consisting only of the July data and pivot the data to create columns for tissue type. Also, the 96 hour timepoint had several samples of poor quality, so we will exclude the the 9g hour timepoint from the final analysis.

```
## # A tibble: 6 x 6
##   intrvl mouse ltr  obs_pt  tissue  pfu_g
##   <dbl> <chr> <chr> <chr> <fct>   <dbl>
## 1     24 1827  A    july_22 Ipsi    2250001
## 2     24 1827  A    july_22 Contra    74075
## 3     24 1827  A    july_22 CLN         1
## 4     24 1827  A    july_22 Kidney    11112
## 5     24 1828  B    july_22 Ipsi    1950196
## 6     24 1828  B    july_22 Contra    66668
```

Create graphs for fig1

We will group by tissue type and split each tissue by time point. Each individual observation is plotted as a dot, with the mean plotted as a bar and the standard error plotted for the error bar.

```
## No summary function supplied, defaulting to 'mean_se()'
## No summary function supplied, defaulting to 'mean_se()'
```



```
## No summary function supplied, defaulting to 'mean_se()'
## No summary function supplied, defaulting to 'mean_se()'
```

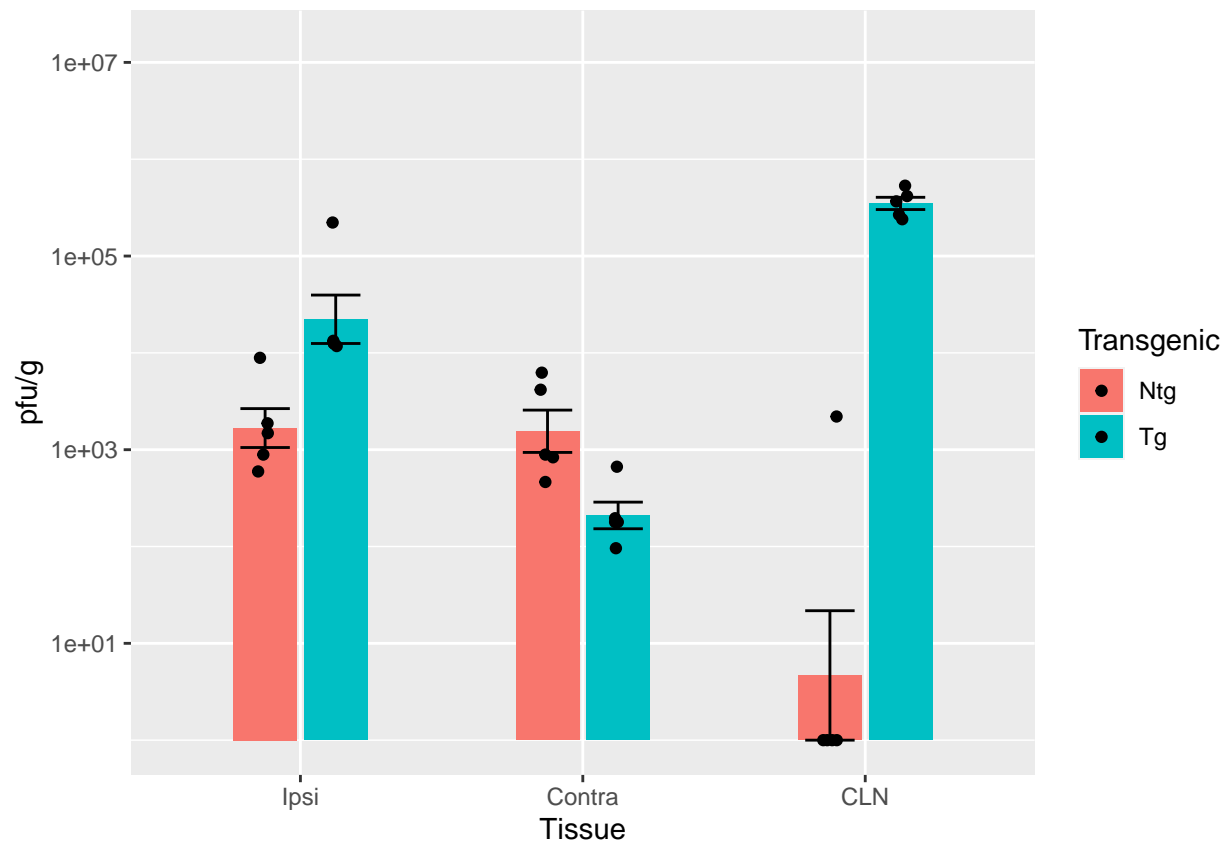
Figure 2 - Toggle experiment August

We will perform a similar analysis for the August time point which is focused on the receptor toggling. We will compare each tissue type by transgenic and non-transgenic mice, so we will create a data set consisting of the August observation point that does include both transgenic and non-transgenic mice.

```
## # A tibble: 6 x 6
##   mouse ltr   intrvl obs_pt   tissue pfu_g
##   <chr> <chr>   <dbl> <chr>   <fct> <dbl>
## 1 Ntg   A         120 august_22 Ipsi    1874
## 2 Ntg   A         120 august_22 Contra    890
## 3 Ntg   A         120 august_22 CLN         1
## 4 Ntg   B         120 august_22 Ipsi    1482
## 5 Ntg   B         120 august_22 Contra    834
## 6 Ntg   B         120 august_22 CLN         1
```

We will graph the August data the same as the July data.

```
## starting httpd help server ... done
## No summary function supplied, defaulting to 'mean_se()'
## No summary function supplied, defaulting to 'mean_se()'
```



```
## No summary function supplied, defaulting to 'mean_se()'
## No summary function supplied, defaulting to 'mean_se()'
```