

Project CAIR: Statistical Analysis of Analytes

Mark Hutchinson, Tory Madden & Gill Bedwell

2025-06-05

Assay panel labelled 1

Load the Data

```
#####  
# ASSAY 1  
#####  
# Load the data from the Excel file  
file_path <- here("R&D3plex_LX200_20240222_1.xlsx")  
data <- read_excel(file_path, sheet = "FI - Bkgd", range = "A62:F156")  
  
## New names:  
## * 'FI - Bkgd' -> 'FI - Bkgd...4'  
## * 'FI - Bkgd' -> 'FI - Bkgd...5'  
## * 'FI - Bkgd' -> 'FI - Bkgd...6'  
  
# Rename columns for easier reference  
colnames(data) <- c("Type", "Well", "Description", "IL1b", "IL6", "TNFa")  
# Fill in missing Description values for the first 14 rows based on Type  
data <- data %>%  
  mutate(Description = ifelse(row_number() <= 14 & is.na(Description), Type, Description))  
# Verify the changes  
head(data, 20)  
  
## # A tibble: 20 x 6  
##   Type Well Description IL1b IL6 TNFa  
##   <chr> <chr> <chr>    <dbl> <dbl> <dbl>  
## 1 B A1 B 7 5 6.5  
## 2 B B1 B 7 4 6  
## 3 S1 C1 S1 6074 2284. 5179.  
## 4 S1 D1 S1 5983 2363 5121.  
## 5 S2 E1 S2 2511 730 2015.  
## 6 S2 F1 S2 2622. 762 2108.  
## 7 S3 G1 S3 846 239 732.  
## 8 S3 H1 S3 894. 240. 744.  
## 9 S4 A2 S4 263 73.5 238.  
## 10 S4 B2 S4 272 78 259.  
## 11 S5 C2 S5 87 25.5 82.8
```

```
## 12 S5    D2    S5          84    26.5    83.8
## 13 S6    E2    S6          27     8.5    27.8
## 14 S6    F2    S6          30     9.5    26.8
## 15 C1    G2    IQC          6    100.    59.8
## 16 C1    H2    IQC          7    97.5    56.8
## 17 X1    A3    CAIR_05     540  1646.    277.
## 18 X1    B3    CAIR_05     556. 1684.    276.
## 19 X2    C3    CAIR_13     434  2166    478.
## 20 X2    D3    CAIR_13     419  2132.    454.
```

```
# Load the expected concentrations from the Excel file
```

```
exp_conc <- read_excel(file_path, sheet = "Exp Conc", range = "D65:F76", col_names = FALSE)
```

```
## New names:
```

```
## * ' ' -> '...1'
```

```
## * ' ' -> '...2'
```

```
## * ' ' -> '...3'
```

```
# Set the correct column names
```

```
colnames(exp_conc) <- c("IL1b", "IL6", "TNFa")
```

```
# Extract the standard data and add expected concentrations
```

```
standard_data <- data %>%
```

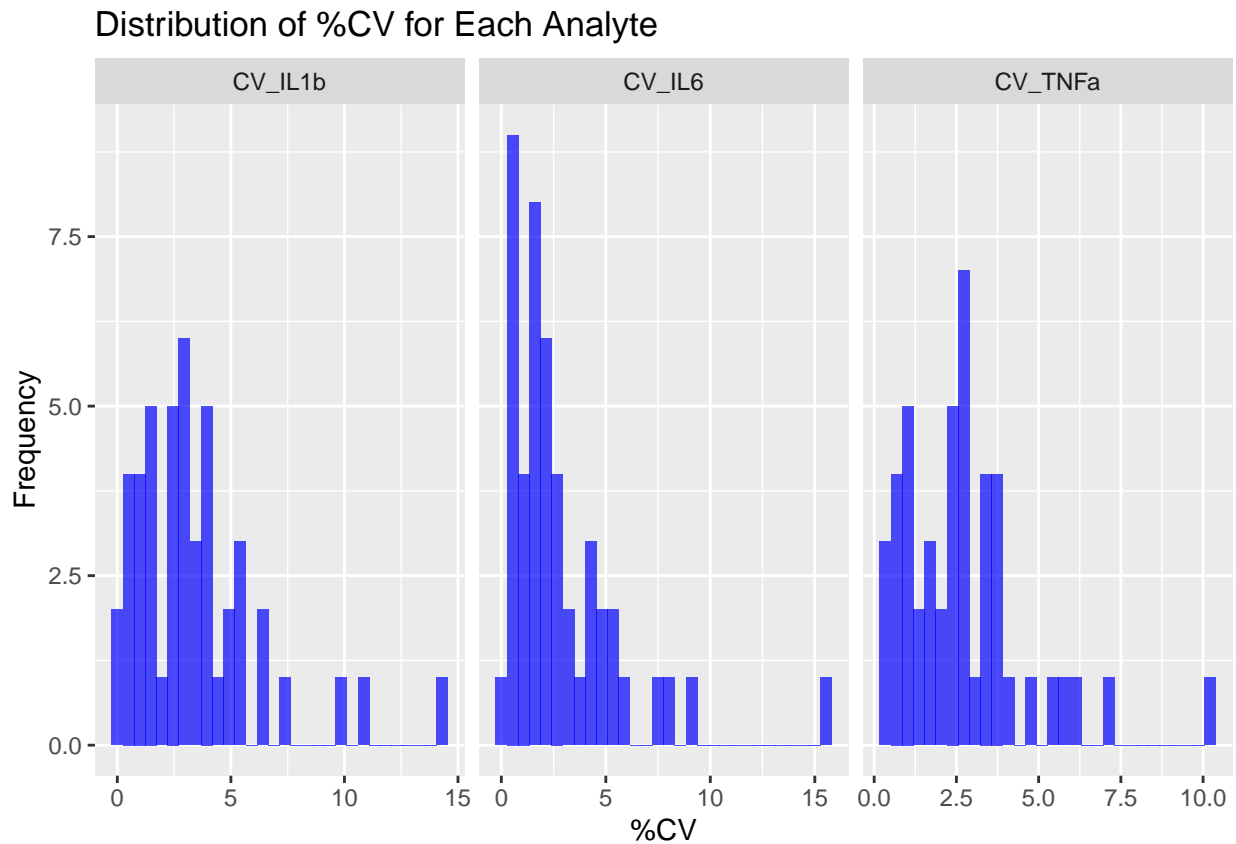
```
  filter(grepl("^S", Type)) %>%
```

```
  mutate(IL1b_exp = exp_conc$IL1b,
```

```
         IL6_exp = exp_conc$IL6,
```

```
         TNFa_exp = exp_conc$TNFa)
```

Calculate Mean and %CV for Replicates



```
## # A tibble: 47 x 4
##   Description CV_IL1b CV_IL6 CV_TNFa
##   <chr>      <dbl> <dbl> <dbl>
## 1 CAIR_60    14.3  9.14  10.1
## 2 IQC        10.9  2.14  3.64
## 3 CAIR_19     9.62  5.45  7.25
## 4 S6         7.44  7.86  2.59
## 5 CAIR_24     6.47  5.29  3.43
## 6 CAIR_105    6.28  2.11  2.52
## 7 CAIR_43     5.67  3.81  2.72
## 8 CAIR_104    5.63  0.786 4.60
## 9 CAIR_14     5.25  2.17  0.431
## 10 CAIR_73    5.11  3.20  0.893
## # i 37 more rows

##
## Call:
## lm(formula = IL1b_exp ~ poly(IL1b, 2), data = standard_data)
##
## Residuals:
##    Min       1Q   Median       3Q      Max
## -63.039 -22.197   0.325  21.051  53.632
##
```

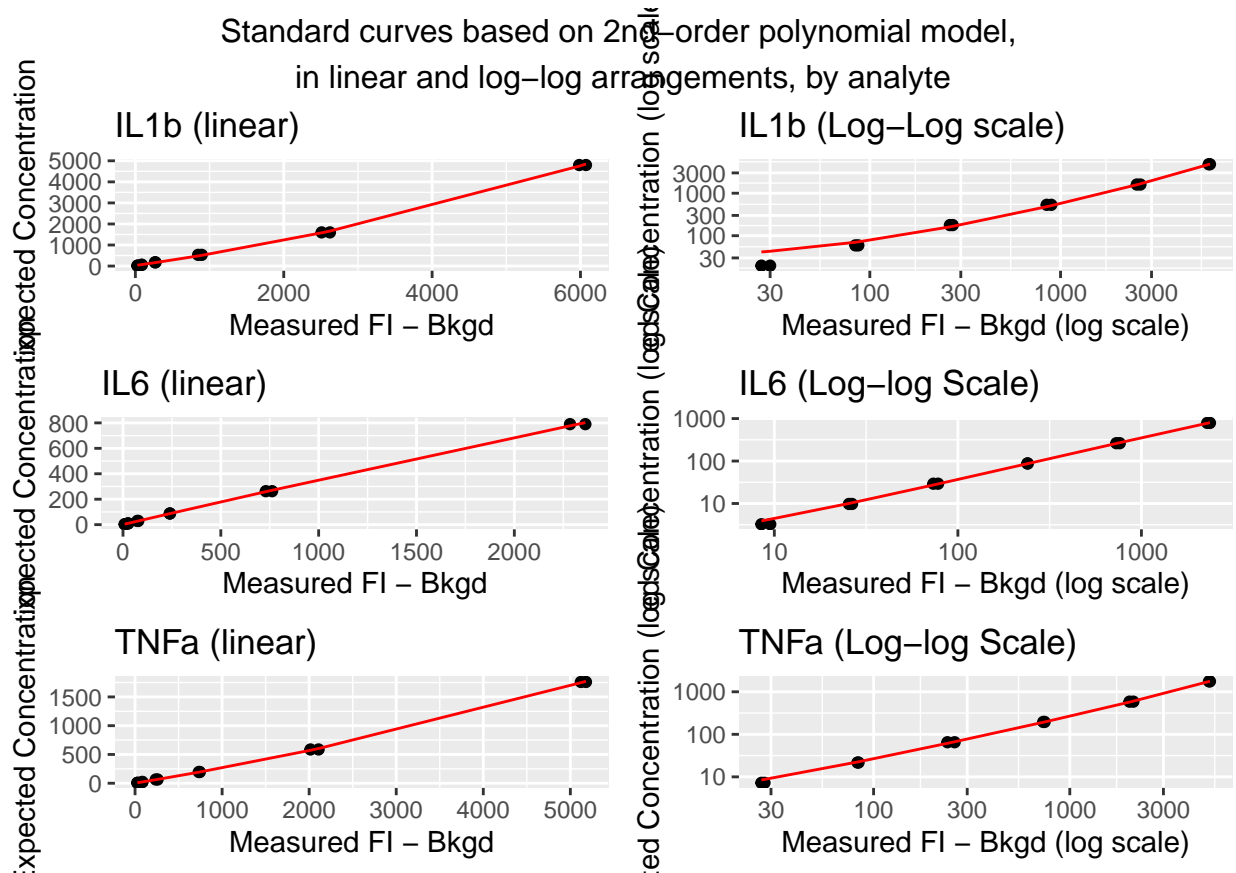
```

## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1198.35      11.47  104.47 3.42e-15 ***
## poly(IL1b, 2)1  5856.39      39.74  147.38 < 2e-16 ***
## poly(IL1b, 2)2   543.90      39.74   13.69 2.49e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 39.74 on 9 degrees of freedom
## Multiple R-squared:  0.9996, Adjusted R-squared:  0.9995
## F-statistic: 1.095e+04 on 2 and 9 DF,  p-value: 5.762e-16

##
## Call:
## lm(formula = IL6_exp ~ poly(IL6, 2), data = standard_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.8997  -0.7412   0.0356   1.4436  13.0685
##
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)    197.228      1.891  104.280 3.48e-15 ***
## poly(IL6, 2)1  967.919      6.552  147.734 < 2e-16 ***
## poly(IL6, 2)2  -13.317      6.552   -2.033  0.0726 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.552 on 9 degrees of freedom
## Multiple R-squared:  0.9996, Adjusted R-squared:  0.9995
## F-statistic: 1.091e+04 on 2 and 9 DF,  p-value: 5.856e-16

##
## Call:
## lm(formula = TNFa_exp ~ poly(TNFa, 2), data = standard_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.2455  -1.5107  -0.7625   3.5380  13.7323
##
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)    439.398      2.742  160.22 < 2e-16 ***
## poly(TNFa, 2)1 2151.909      9.500  226.52 < 2e-16 ***
## poly(TNFa, 2)2  145.681      9.500   15.34 9.31e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.5 on 9 degrees of freedom
## Multiple R-squared:  0.9998, Adjusted R-squared:  0.9998
## F-statistic: 2.577e+04 on 2 and 9 DF,  p-value: < 2.2e-16

```



```
## # A tibble: 12 x 11
##   Type Well IL1b_exp Fitted_IL1b CV_IL1b_fitted IL6_exp Fitted_IL6
##   <chr> <chr>   <dbl>      <dbl>      <dbl>    <dbl>      <dbl>
## 1 S1    C1      4800      4845.        0.935    790       777.
## 2 S1    D1      4800      4746.        1.13     790       802.
## 3 S2    E1      1600      1580.        1.28     263.      259.
## 4 S2    F1      1600      1663.        3.79     263.      270.
## 5 S3    G1       533.      482.        10.7     87.8      86.5
## 6 S3    H1       533.      510.         4.61     87.8      87.0
## 7 S4    A2       178.      162.         9.99     29.3      27.3
## 8 S4    B2       178.      166.         6.89     29.3      28.9
## 9 S5    C2        59.3     71.6        17.2     9.75     10.0
## 10 S5   D2        59.3     70.1        15.4     9.75     10.4
## 11 S6   E2        19.8     41.6        52.5     3.25     3.90
## 12 S6   F2        19.8     43.1        54.1     3.25     4.26
## # i 4 more variables: CV_IL6_fitted <dbl>, TNFa_exp <dbl>, Fitted_TNFa <dbl>,
## #   CV_TNFa_fitted <dbl>

## Warning in styling_latex_scale(out, table_info, "down"): Longtable cannot be
## resized.
```

Table 1: Increasing CV values with decreasing expected concentration

Type	Well	IL1b_exp	CV_IL1b_fitted	IL6_exp	CV_IL6_fitted	TNFa_exp	CV_TNFa_fitted
S1	C1	4800.00	0.9349825	790.00	1.6820623	1760.00	0.6967211
S1	D1	4800.00	1.1299642	790.00	1.4839377	1760.00	0.7543601
S2	E1	1600.00	1.2807619	263.33	1.5802649	586.67	2.3968155
S2	F1	1600.00	3.7906003	263.33	2.5919339	586.67	2.6944934
S3	G1	533.33	10.7194518	87.78	1.4678301	195.56	1.8078734
S3	H1	533.33	4.6100689	87.78	0.8448837	195.56	0.1676885
S4	A2	177.78	9.9894885	29.26	7.1988377	65.19	6.0756277
S4	B2	177.78	6.8899609	29.26	1.2043863	65.19	2.4396745
S5	C2	59.26	17.2096369	9.75	2.7623629	21.73	2.8514815
S5	D2	59.26	15.4271226	9.75	6.1341502	21.73	3.9224877
S6	E2	19.75	52.4941609	3.25	16.6895767	7.24	16.8856669
S6	F2	19.75	54.1397922	3.25	23.7369677	7.24	14.4573864

Interpretation: VERY poor fit at lower end of expected concentrations.

Use weighting strategy to address poor fit at lower end of concentration range

```
##
## Call:
## lm(formula = IL1b_exp ~ poly(IL1b, 2), data = standard_data,
##     weights = weight_IL1b)
##
## Weighted Residuals:
##      Min       1Q   Median       3Q      Max
## -2.7653 -0.6808  0.4037  1.2363  1.8214
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1196.68     14.90  80.302 3.65e-14 ***
## poly(IL1b, 2)1   5847.84     87.30  66.989 1.86e-13 ***
## poly(IL1b, 2)2    455.14     50.68   8.981 8.69e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.492 on 9 degrees of freedom
## Multiple R-squared:  0.9985, Adjusted R-squared:  0.9982
## F-statistic: 3011 on 2 and 9 DF, p-value: 1.916e-13

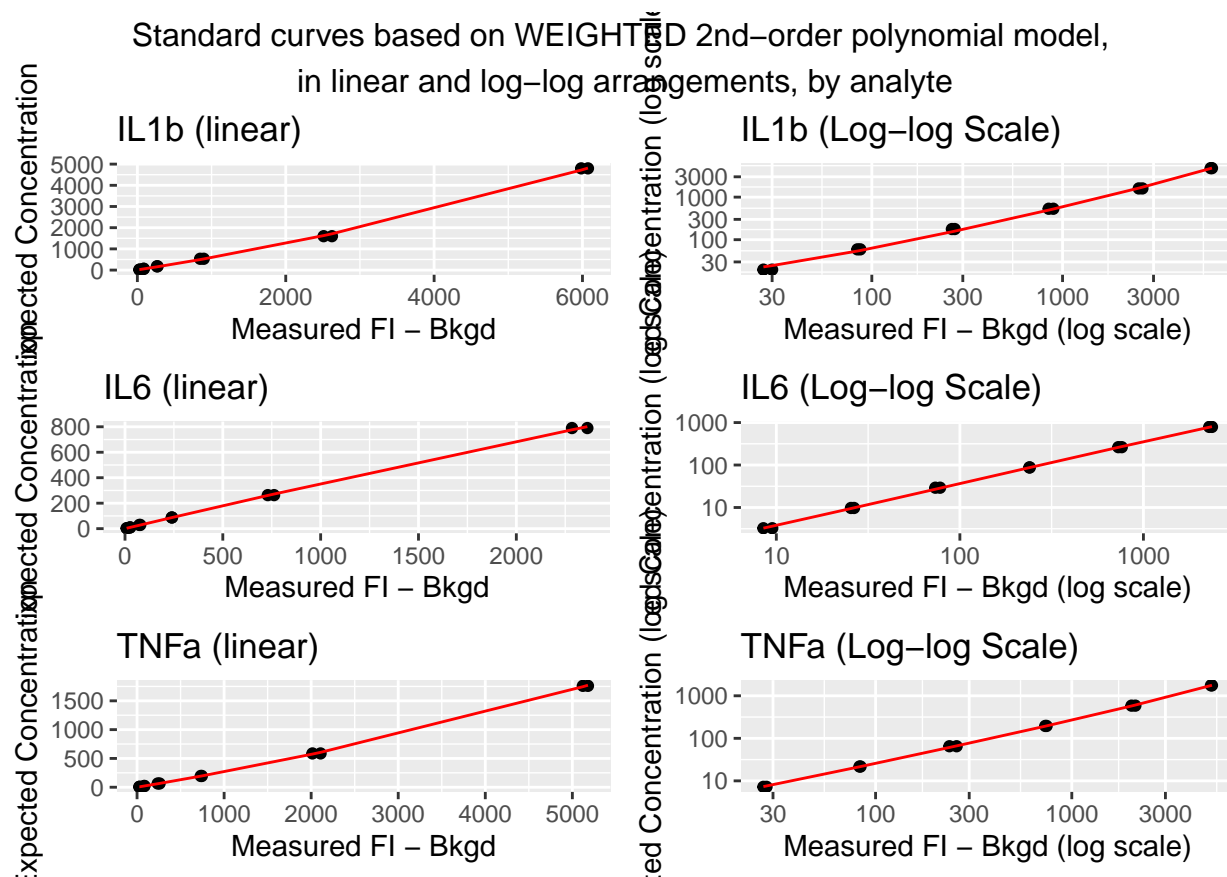
##
## Call:
## lm(formula = IL6_exp ~ poly(IL6, 2), data = standard_data, weights = weight_IL6)
##
## Weighted Residuals:
##      Min       1Q   Median       3Q      Max
## -0.52694 -0.07176  0.07097  0.13621  0.48419
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
```

```

## (Intercept)    197.148      1.325 148.846 < 2e-16 ***
## poly(IL6, 2)1  967.752      8.030 120.523 9.47e-16 ***
## poly(IL6, 2)2 -16.038      4.397  -3.647  0.00534 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3268 on 9 degrees of freedom
## Multiple R-squared:  0.9996, Adjusted R-squared:  0.9995
## F-statistic: 1.035e+04 on 2 and 9 DF,  p-value: 7.45e-16

##
## Call:
## lm(formula = TNFa_exp ~ poly(TNFa, 2), data = standard_data,
##     weights = weight_TNFa)
##
## Weighted Residuals:
##      Min       1Q   Median       3Q      Max
## -0.78286 -0.13576 -0.01197  0.23114  0.52182
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    439.285      2.356   186.5 < 2e-16 ***
## poly(TNFa, 2)1 2151.353     13.921   154.5 < 2e-16 ***
## poly(TNFa, 2)2  140.615      7.768    18.1 2.18e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3894 on 9 degrees of freedom
## Multiple R-squared:  0.9997, Adjusted R-squared:  0.9997
## F-statistic: 1.624e+04 on 2 and 9 DF,  p-value: < 2.2e-16

```



Back-calculate standard curve observed values with weighted model

```
## # A tibble: 12 x 11
##   Type Well IL1b_exp Fitted_IL1b_weighted CV_IL1b_weighted IL6_exp
##   <chr> <chr>   <dbl>           <dbl>           <dbl>   <dbl>
## 1 S1    C1      4800             4813.           0.280   790
## 2 S1    D1      4800             4719.           1.71    790
## 3 S2    E1      1600             1627.           1.64   263.
## 4 S2    F1      1600             1711.           6.47   263.
## 5 S3    G1       533.             496.           7.61   87.8
## 6 S3    H1       533.             525.           1.52   87.8
## 7 S4    A2       178.             153.          15.8   29.3
## 8 S4    B2       178.             159.          12.1   29.3
## 9 S5    C2        59.3           55.7           6.39   9.75
## 10 S5   D2        59.3           54.1           9.62   9.75
## 11 S6    E2        19.8           22.9          13.9   3.25
## 12 S6    F2        19.8           24.6          19.6   3.25
## # i 5 more variables: Fitted_IL6_weighted <dbl>, CV_IL6_weighted <dbl>,
## #   TNFa_exp <dbl>, Fitted_TNFa_weighted <dbl>, CV_TNFa_weighted <dbl>

## Warning in styling_latex_scale(out, table_info, "down"): Longtable cannot be
## resized.
```

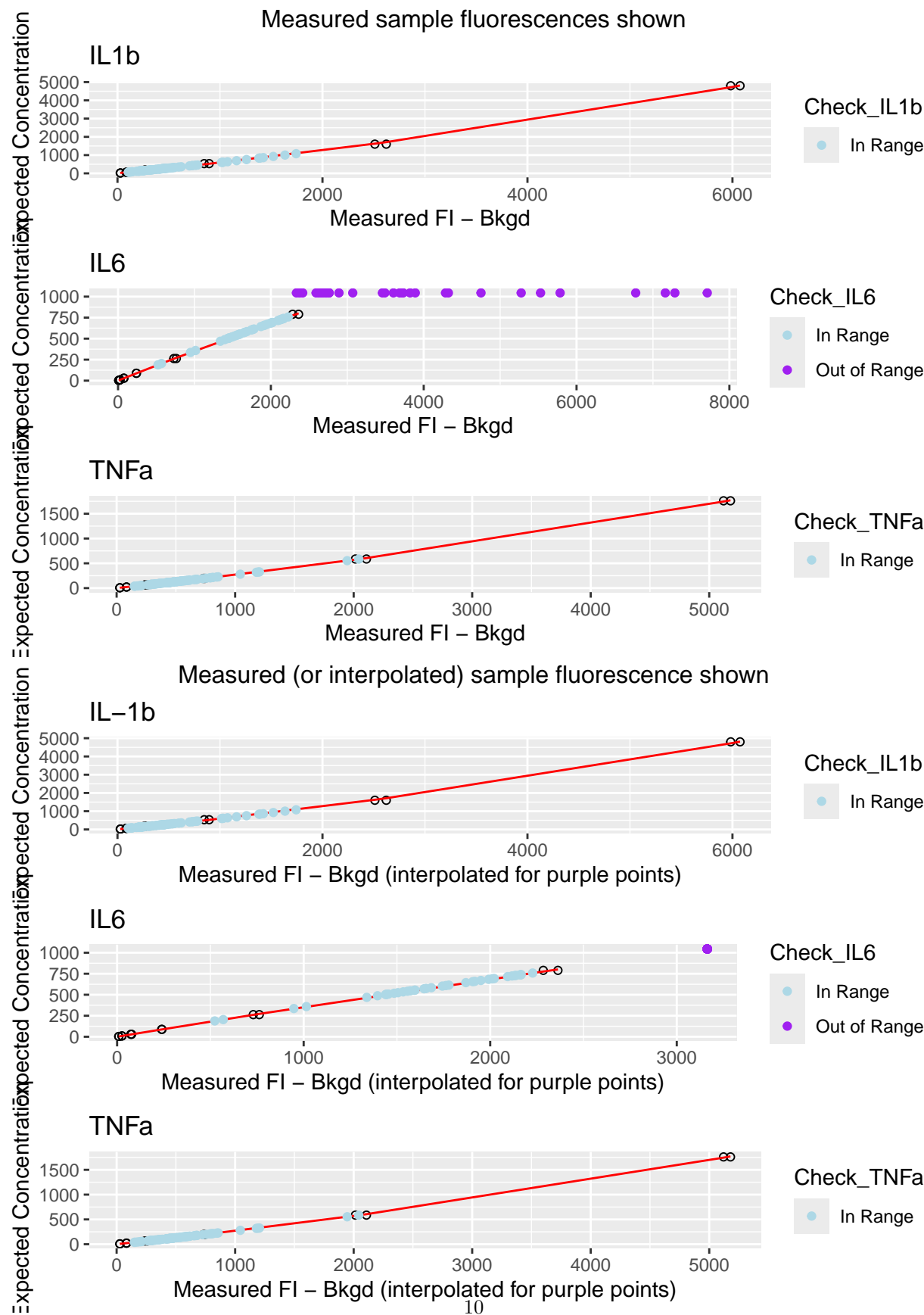

Table 2: Less dramatic increase in CV values with decreasing expected concentration

Type	Well	IL1b_exp	CV_IL1b_weighted	IL6_exp	CV_IL6_weighted	TNFa_exp	CV_TNFa_weighted
S1	C1	4800.00	0.2797202	790.00	1.7528756	1760.00	0.5972466
S1	D1	4800.00	1.7148201	790.00	1.3752796	1760.00	0.8452870
S2	E1	1600.00	1.6424497	263.33	0.9979349	586.67	1.9230086
S2	F1	1600.00	6.4662283	263.33	3.1451082	586.67	3.1309133
S3	G1	533.33	7.6064360	87.78	1.1670563	195.56	1.3844900
S3	H1	533.33	1.5184838	87.78	0.5418493	195.56	0.2554541
S4	A2	177.78	15.8221186	29.26	8.6450569	65.19	6.9094498
S4	B2	177.78	12.1193517	29.26	2.4270113	65.19	1.8247720
S5	C2	59.26	6.3857884	9.75	3.0166959	21.73	1.4367081
S5	D2	59.26	9.6227409	9.75	0.8027859	21.73	0.2552600
S6	E2	19.75	13.9342562	3.25	0.4890223	7.24	4.6419982
S6	F2	19.75	19.6445872	3.25	10.4872180	7.24	1.3891033

Predict values for unknown samples

```
## # A tibble: 78 x 12
##   Type Well Description IL1b IL6 TNFa Predicted_IL1b Predicted_IL6
##   <chr> <chr> <chr>    <dbl> <dbl> <dbl>         <dbl>         <dbl>
## 1 X1    A3    CAIR_05    540 1646. 277.         313.         571.
## 2 X1    B3    CAIR_05    556. 1684. 276.         322.         584.
## 3 X2    C3    CAIR_13    434 2166 478.         251.         739.
## 4 X2    D3    CAIR_13    419 2132. 454.         242.         728.
## 5 X4    G3    CAIR_20    434. 2416. 677.         251.         818.
## 6 X4    H3    CAIR_20    412. 2334. 653.         238.         792.
## 7 X5    A4    CAIR_44    450 2648. 477.         260.         889.
## 8 X5    B4    CAIR_44    427 2592. 460.         247.         872.
## 9 X6    C4    CAIR_42   1030 2750. 814.         609.         920.
## 10 X6   D4    CAIR_42   1026. 2675 789.         607.         898.
## # i 68 more rows
## # i 4 more variables: Predicted_TNFa <dbl>, Check_IL1b <chr>, Check_IL6 <chr>,
## #   Check_TNFa <chr>
```

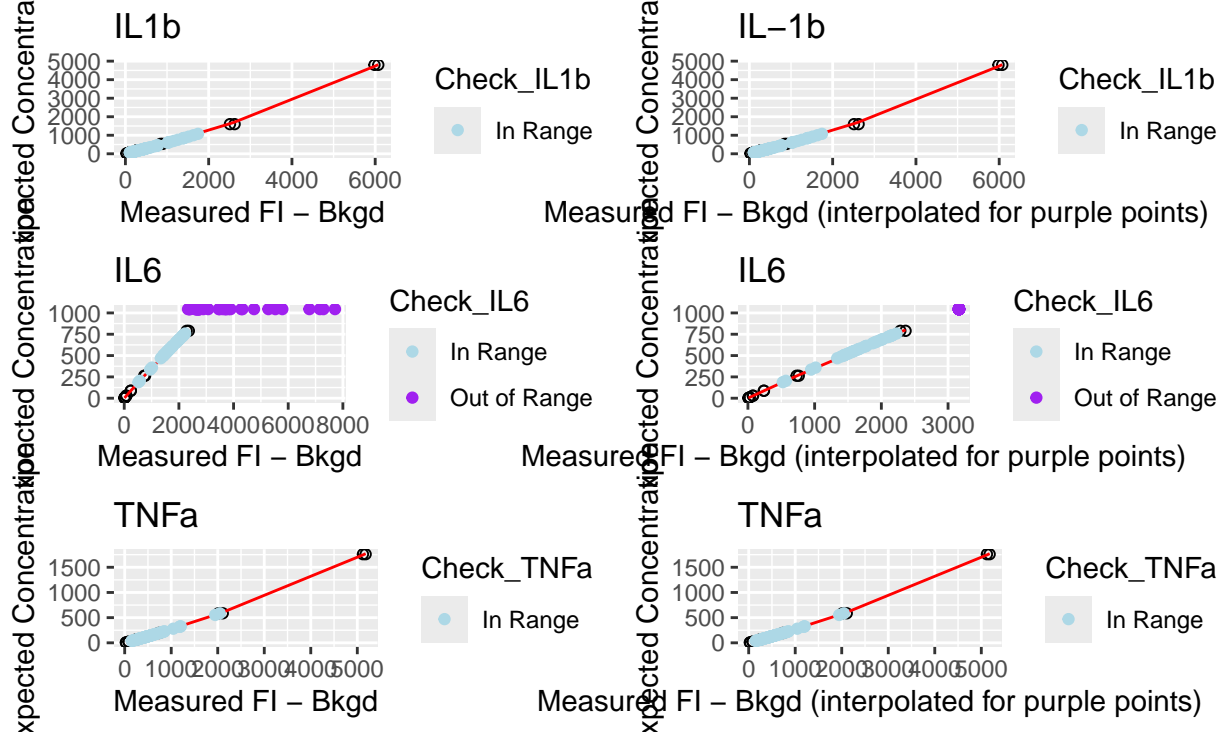
Interpolate values for samples flagged as “out of range”



Weighted Standard Curve (red) and data points. Black: standards.

Blue: sample values in range. Purple: values interpolated for out of range samples

Measured sample fluorescences shown (or interpolated) sample fluorescence s



```
## TableGrob (2 x 2) "arrange": 3 grobs
##   z   cells   name      grob
## 1 1 (2-2,1-1) arrange  gtable[arrange]
## 2 2 (2-2,2-2) arrange  gtable[arrange]
## 3 3 (1-1,1-2) arrange text[GRID.text.952]
```

Calculate means and multiply by 30

```
# Calculate the mean of the duplicates based on the Description value and include the check value
mean_duplicates <- reportable_data %>%
  group_by(Description) %>%
  summarise(Mean_IL1b = (mean(report_IL1b))*30, #calculate mean and multiply by 30 to account for dilut
            Mean_IL6 = (mean(report_IL6))*30,
            Mean_TNFa = (mean(report_TNFa))*30,
            Check_IL1b = ifelse(any(Check_IL1b == "Out of Range"), "Out of Range", "In Range"),
            Check_IL6 = ifelse(any(Check_IL6 == "Out of Range"), "Out of Range", "In Range"),
            Check_TNFa = ifelse(any(Check_TNFa == "Out of Range"), "Out of Range", "In Range"))

# Display the mean of the duplicates
# mean_duplicates
```

Save results

```
# Save the unknown raw values to a CSV file
write.csv(unknown_data, here("unknown_raw_values.csv"), row.names = FALSE)

# Save the mean of the duplicates to a CSV file
readr::write_rds(mean_duplicates, here("luminex_results.rds"))
write.csv(mean_duplicates, here("luminex_results.csv"), row.names = FALSE)
```

Assay panel labelled 2

Load the Data

```
# Load the data from the Excel file
#####
# ASSAY 2
#####
file_path_2 <- here("R&D3plex_LX200_20240222_2.xlsx")
data_2 <- read_excel(file_path_2, sheet = "FI - Bkgd", range = "A61:F153")

## New names:
## * 'FI - Bkgd' -> 'FI - Bkgd...4'
## * 'FI - Bkgd' -> 'FI - Bkgd...5'
## * 'FI - Bkgd' -> 'FI - Bkgd...6'

# Rename columns for easier reference
colnames(data_2) <- c("Type", "Well", "Description", "IL1b", "IL6", "TNFa")
# Fill in missing Description values for the first 14 rows based on Type
data_2 <- data_2 %>%
  mutate(Description = ifelse(row_number() <= 14 & is.na(Description), Type, Description))
# Verify the changes
head(data_2, 20)
```

```
## # A tibble: 20 x 6
##   Type Well Description  IL1b    IL6  TNFa
##   <chr> <chr> <chr>      <dbl> <dbl> <dbl>
## 1 B     A1     B         7.5    5     7.5
## 2 B     B1     B          7     5     6
## 3 S1    C1    S1      6736. 2617  5655.
## 4 S1    D1    S1      6576. 2722. 5667.
## 5 S2    E1    S2      2862.  876. 2373.
## 6 S2    F1    S2      2915.  885  2371.
## 7 S3    G1    S3       970.  270   819.
## 8 S3    H1    S3       956.  280.  849.
## 9 S4    A2    S4       274.   85   262.
## 10 S4   B2    S4       289.  88.5  271.
## 11 S5    C2    S5       87.8   28   91.3
## 12 S5    D2    S5       97.3   29   90.8
```

```
## 13 S6      E2      S6          31.8    9.5    29.8
## 14 S6      F2      S6          31.8     9    30.3
## 15 C1      G2      IQC          7.3   110    64.3
## 16 C1      H2      IQC          7.8   110.    66.3
## 17 X1      A3      CAIR_86      430.  1934.   410.
## 18 X1      B3      CAIR_86      427.  1962    421.
## 19 X2      C3      CAIR_47      328.  3246.   474.
## 20 X2      D3      CAIR_47      305.  3066    466.
```

```
# Load the expected concentrations from the Excel file
exp_conc_2 <- read_excel(file_path_2, sheet = "Exp Conc", range = "D64:F75", col_names = FALSE)
```

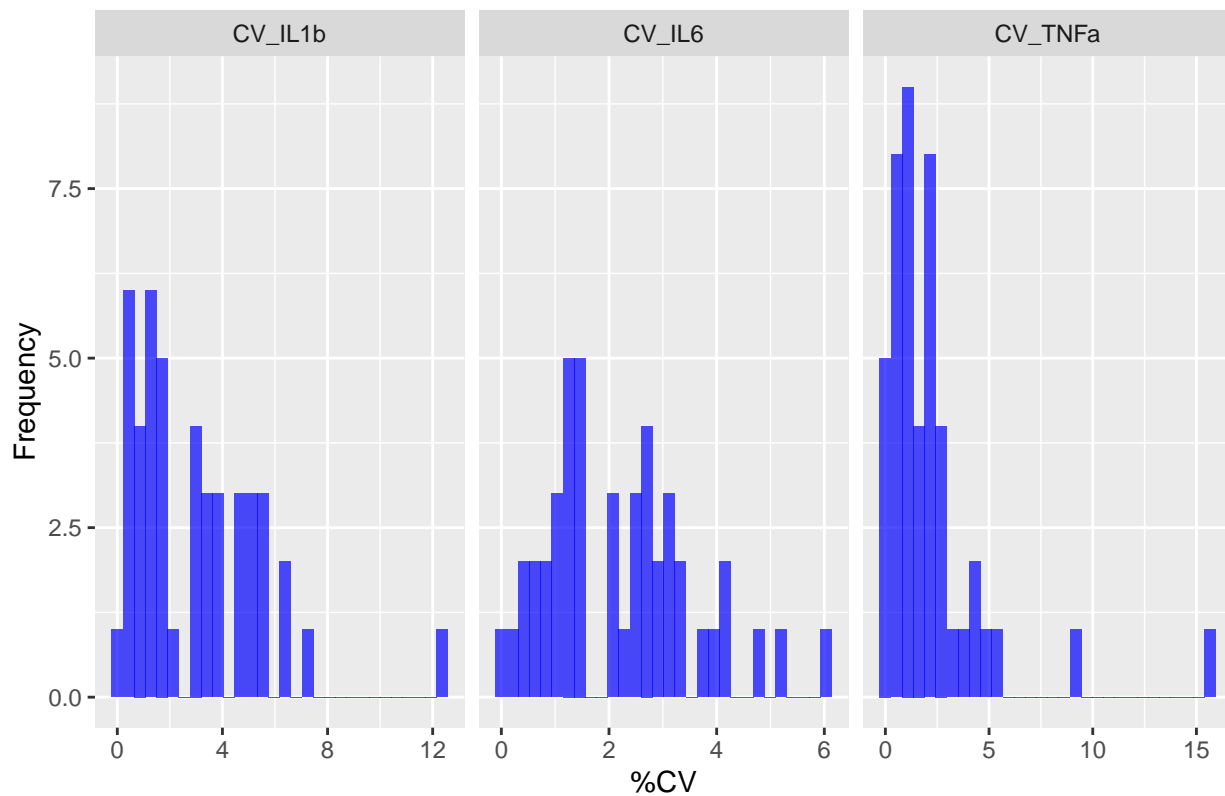
```
## New names:
## * ' ' -> '...1'
## * ' ' -> '...2'
## * ' ' -> '...3'
```

```
# Set the correct column names
colnames(exp_conc_2) <- c("IL1b", "IL6", "TNFa")

# Extract the standard data and add expected concentrations
standard_data_2 <- data_2 %>%
  filter(grepl("^S", Type)) %>%
  mutate(IL1b_exp = exp_conc_2$IL1b,
         IL6_exp = exp_conc_2$IL6,
         TNFa_exp = exp_conc_2$TNFa)
```

Calculate Mean and %CV for Replicates

Distribution of %CV for Each Analyte



```
## # A tibble: 46 x 4
##   Description CV_IL1b CV_IL6 CV_TNFa
##   <chr>         <dbl> <dbl> <dbl>
## 1 CAIR_205      12.4   6.03  9.36
## 2 S5            7.26   2.48  0.388
## 3 CAIR_223      6.60   4.09  0.743
## 4 CAIR_232      6.45   1.42  4.10
## 5 CAIR_231      5.67   2.70  3.58
## 6 CAIR_202      5.64   3.31  5.34
## 7 CAIR_252      5.52   2.18  2.06
## 8 CAIR_47       5.13   4.02  1.20
## 9 CAIR_236      5.03   5.11  1.36
## 10 CAIR_240     5.03   2.40  0.972
## # i 36 more rows

##
## Call:
## lm(formula = IL1b_exp ~ poly(IL1b, 2), data = standard_data_2)
##
## Residuals:
##    Min     1Q  Median     3Q    Max
## -73.47 -28.46 -12.21  28.11  86.45
##
```

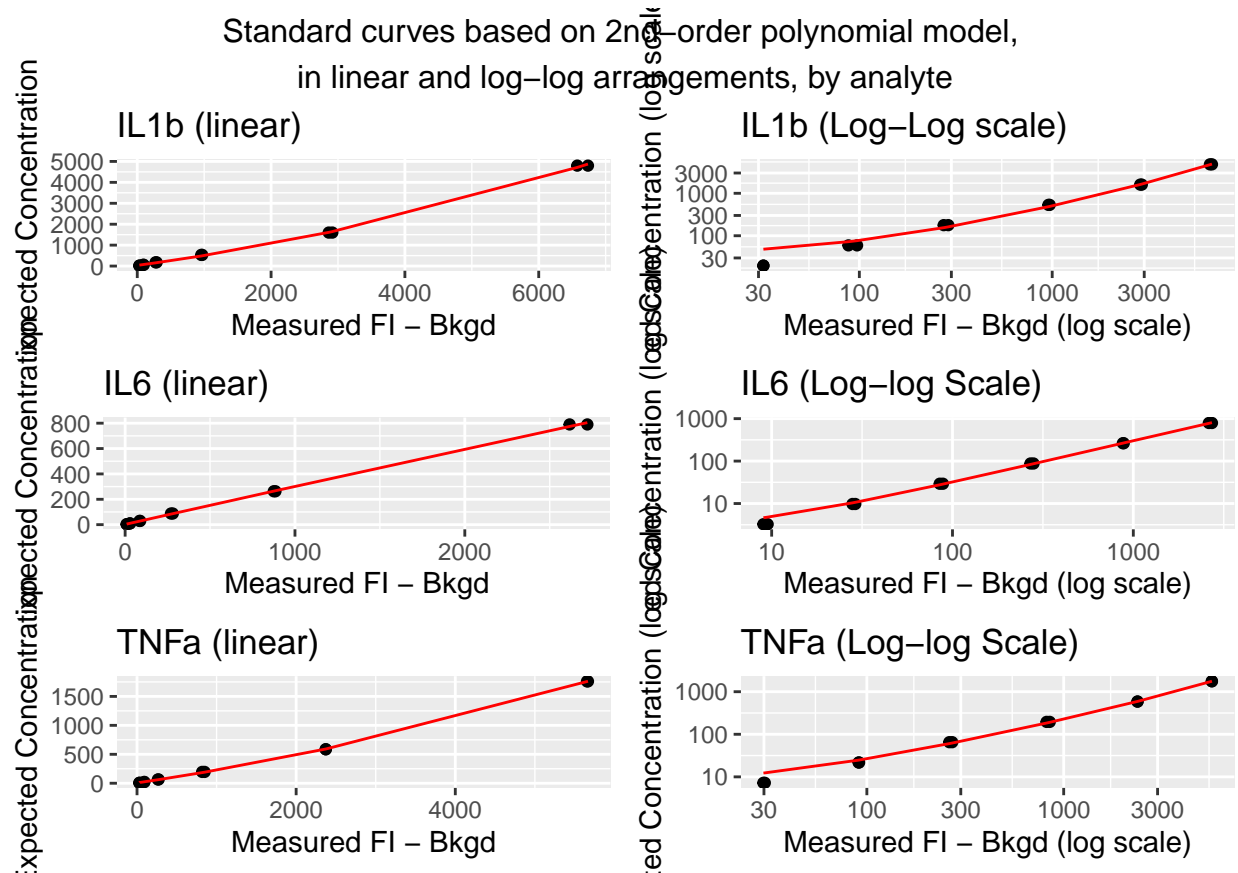
```

## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1198.35      14.46   82.87 2.75e-14 ***
## poly(IL1b, 2)1  5851.51      50.09  116.81 1.25e-15 ***
## poly(IL1b, 2)2   586.95      50.09   11.72 9.43e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 50.09 on 9 degrees of freedom
## Multiple R-squared:  0.9993, Adjusted R-squared:  0.9992
## F-statistic: 6891 on 2 and 9 DF, p-value: 4.634e-15

##
## Call:
## lm(formula = IL6_exp ~ poly(IL6, 2), data = standard_data_2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.1776  -1.4145  -0.7718   1.3680  15.8617
##
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)    197.228      2.152  91.663 1.11e-14 ***
## poly(IL6, 2)1  967.937      7.454 129.862 4.84e-16 ***
## poly(IL6, 2)2   -5.463      7.454  -0.733   0.482
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.454 on 9 degrees of freedom
## Multiple R-squared:  0.9995, Adjusted R-squared:  0.9993
## F-statistic: 8432 on 2 and 9 DF, p-value: 1.869e-15

##
## Call:
## lm(formula = TNFa_exp ~ poly(TNFa, 2), data = standard_data_2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -5.143  -4.478  -2.418   3.519  11.755
##
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)    439.398      1.717  255.97 < 2e-16 ***
## poly(TNFa, 2)1 2148.922      5.946  361.38 < 2e-16 ***
## poly(TNFa, 2)2  185.916      5.946   31.27 1.72e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.946 on 9 degrees of freedom
## Multiple R-squared:  0.9999, Adjusted R-squared:  0.9999
## F-statistic: 6.579e+04 on 2 and 9 DF, p-value: < 2.2e-16

```



```
## # A tibble: 12 x 11
##   Type Well IL1b_exp Fitted_IL1b CV_IL1b_fitted IL6_exp Fitted_IL6
##   <chr> <chr>   <dbl>      <dbl>      <dbl>      <dbl>      <dbl>
## 1 S1    C1      4800      4873.        1.51      790      774.
## 2 S1    D1      4800      4714.        1.83      790      804.
## 3 S2    E1      1600      1611.        0.698     263.     265.
## 4 S2    F1      1600      1647.        2.88     263.     267.
## 5 S3    G1       533.       490.        8.88     87.8     83.4
## 6 S3    H1       533.       483.       10.5     87.8     86.5
## 7 S4    A2       178.       155.       14.8     29.3     27.6
## 8 S4    B2       178.       162.        9.88     29.3     28.6
## 9 S5    C2        59.3      72.4       18.2     9.75     10.4
## 10 S5   D2        59.3      76.6       22.6     9.75     10.7
## 11 S6    E2        19.8      48.2       59.0     3.25     4.78
## 12 S6    F2        19.8      48.2       59.0     3.25     4.63
## # i 4 more variables: CV_IL6_fitted <dbl>, TNFa_exp <dbl>, Fitted_TNFa <dbl>,
## #   CV_TNFa_fitted <dbl>

## Warning in styling_latex_scale(out, table_info, "down"): Longtable cannot be
## resized.
```

Table 3: Increasing CV values with decreasing expected concentration

Type	Well	IL1b_exp	CV_IL1b_fitted	IL6_exp	CV_IL6_fitted	TNFa_exp	CV_TNFa_fitted
S1	C1	4800.00	1.5074925	790.00	2.0489487	1760.00	0.1789305
S1	D1	4800.00	1.8341263	790.00	1.7629914	1760.00	0.1064136
S2	E1	1600.00	0.6984172	263.33	0.4510414	586.67	0.8309207
S2	F1	1600.00	2.8813800	263.33	1.5033525	586.67	0.7328970
S3	G1	533.33	8.8814991	87.78	5.3106100	195.56	6.3954796
S3	H1	533.33	10.4937518	87.78	1.4649864	195.56	2.4332955
S4	A2	177.78	14.8462294	29.26	6.0514606	65.19	8.0286960
S4	B2	177.78	9.8778827	29.26	2.1393563	65.19	4.7275020
S5	C2	59.26	18.1764060	9.75	5.9849439	21.73	12.3568806
S5	D2	59.26	22.5955523	9.75	8.6474446	21.73	11.9937371
S6	E2	19.75	59.0325654	3.25	31.9787046	7.24	41.0512726
S6	F2	19.75	59.0325654	3.25	29.7560514	7.24	41.5327347

Interpretation: VERY poor fit at lower end of expected concentrations.

Use weighting strategy to address poor fit at lower end of concentration range

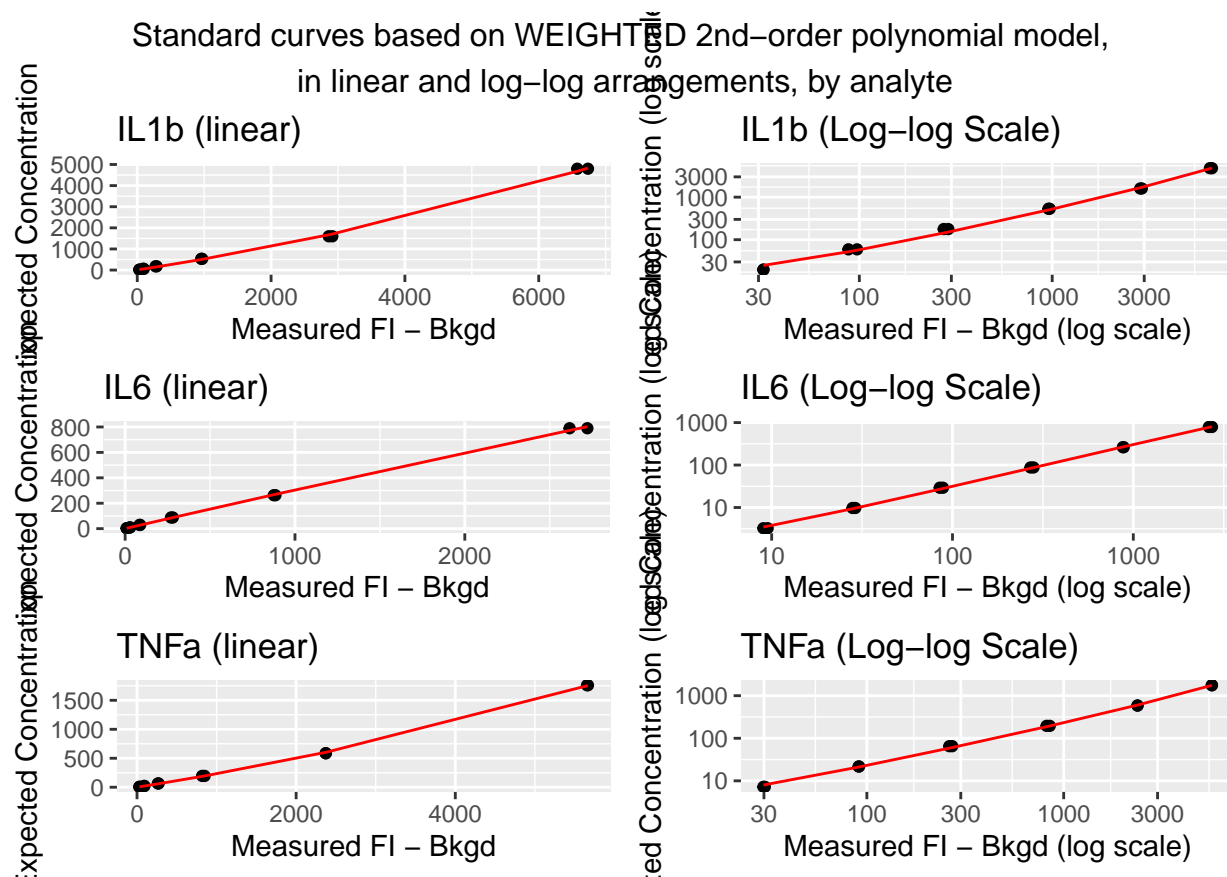
```
##
## Call:
## lm(formula = IL1b_exp ~ poly(IL1b, 2), data = standard_data_2,
##     weights = weight_IL1b)
##
## Weighted Residuals:
##      Min       1Q   Median       3Q      Max
## -2.6663 -1.1596  0.6391  1.4917  2.5563
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1195.80      18.43   64.891 2.47e-13 ***
## poly(IL1b, 2)1   5840.21     107.66   54.246 1.24e-12 ***
## poly(IL1b, 2)2    474.78       63.44    7.484 3.75e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.846 on 9 degrees of freedom
## Multiple R-squared:  0.9977, Adjusted R-squared:  0.9972
## F-statistic: 1966 on 2 and 9 DF, p-value: 1.299e-12
##
## Call:
## lm(formula = IL6_exp ~ poly(IL6, 2), data = standard_data_2,
##     weights = weight_IL6)
##
## Weighted Residuals:
##      Min       1Q   Median       3Q      Max
## -0.44651 -0.22181  0.05319  0.27563  0.59621
##
## Coefficients:
```

```

##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    197.120      1.538 128.206 5.43e-16 ***
## poly(IL6, 2)1  967.714      9.298 104.076 3.54e-15 ***
## poly(IL6, 2)2 -10.564      5.157  -2.048  0.0708 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3794 on 9 degrees of freedom
## Multiple R-squared:  0.9994, Adjusted R-squared:  0.9993
## F-statistic: 7676 on 2 and 9 DF, p-value: 2.853e-15

##
## Call:
## lm(formula = TNFa_exp ~ poly(TNFa, 2), data = standard_data_2,
##     weights = weight_TNFa)
##
## Weighted Residuals:
##      Min       1Q   Median       3Q      Max
## -0.6629 -0.2917  0.1112   0.3118   0.8122
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)      439.19       3.16  139.00 2.62e-16 ***
## poly(TNFa, 2)1  2147.14      18.55  115.77 1.36e-15 ***
## poly(TNFa, 2)2   165.36      10.58   15.63 7.90e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5223 on 9 degrees of freedom
## Multiple R-squared:  0.9995, Adjusted R-squared:  0.9994
## F-statistic: 9022 on 2 and 9 DF, p-value: 1.379e-15

```



Back-calculate standard curve observed values with weighted model

```
## # A tibble: 12 x 11
##   Type Well IL1b_exp Fitted_IL1b_weighted CV_IL1b_weighted IL6_exp
##   <chr> <chr>   <dbl>           <dbl>           <dbl>   <dbl>
## 1 S1    C1      4800            4830.            0.629   790
## 2 S1    D1      4800            4680.            2.57    790
## 3 S2    E1      1600            1670.            4.20    263.
## 4 S2    F1      1600            1707.            6.25    263.
## 5 S3    G1       533.            508.            4.89    87.8
## 6 S3    H1       533.            501.            6.50    87.8
## 7 S4    A2       178.            144.            23.7    29.3
## 8 S4    B2       178.            151.            17.4    29.3
## 9 S5    C2        59.3         52.0            13.9     9.75
## 10 S5   D2        59.3         56.7             4.60     9.75
## 11 S6    E2        19.8         24.9            20.7     3.25
## 12 S6    F2        19.8         24.9            20.7     3.25
## # i 5 more variables: Fitted_IL6_weighted <dbl>, CV_IL6_weighted <dbl>,
## #   TNFa_exp <dbl>, Fitted_TNFa_weighted <dbl>, CV_TNFa_weighted <dbl>

## Warning in styling_latex_scale(out, table_info, "down"): Longtable cannot be
## resized.
```

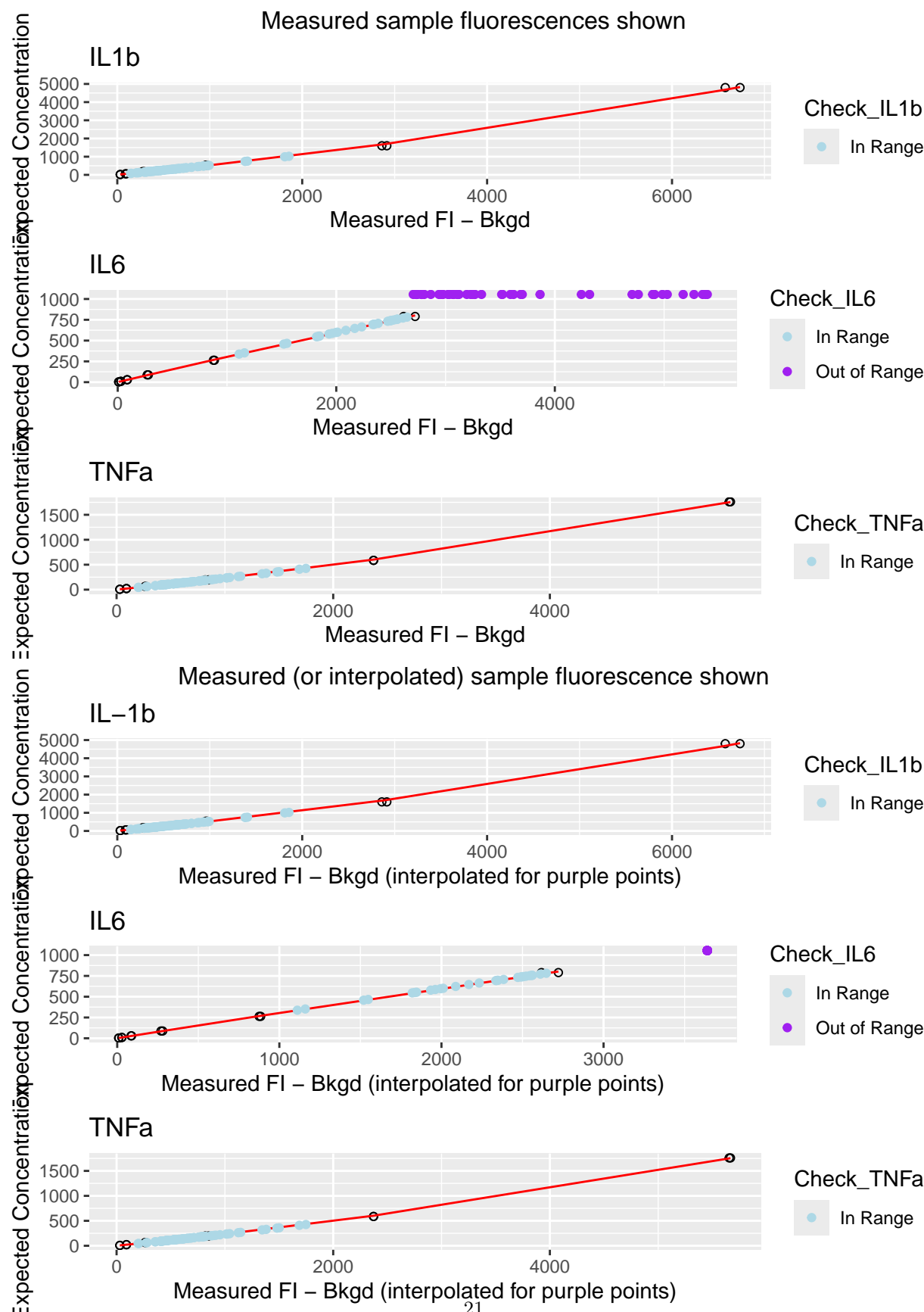
Table 4: Less dramatic increase in CV values with decreasing expected concentration

Type	Well	IL1b_exp	CV_IL1b_weighted	IL6_exp	CV_IL6_weighted	TNFa_exp	CV_TNFa_weighted
S1	C1	4800.00	0.6286797	790.00	2.1671731	1760.00	0.5450275
S1	D1	4800.00	2.5741493	790.00	1.5637815	1760.00	0.2660460
S2	E1	1600.00	4.2005198	263.33	1.5246932	586.67	2.6640785
S2	F1	1600.00	6.2491709	263.33	2.5626368	586.67	2.5691234
S3	G1	533.33	4.8940516	87.78	4.7085932	195.56	4.4019225
S3	H1	533.33	6.4997245	87.78	0.8387264	195.56	0.4558821
S4	A2	177.78	23.7196056	29.26	8.5627272	65.19	11.1845947
S4	B2	177.78	17.3898820	29.26	4.3817498	65.19	7.5222445
S5	C2	59.26	13.8994120	9.75	4.2550523	21.73	2.4402967
S5	D2	59.26	4.6041275	9.75	0.9197482	21.73	2.9642962
S6	E2	19.75	20.6933884	3.25	10.5221521	7.24	9.4791566
S6	F2	19.75	20.6933884	3.25	6.5431596	7.24	10.6731994

Predict values for unknown samples

```
## # A tibble: 76 x 12
##   Type Well Description IL1b IL6 TNFa Predicted_IL1b Predicted_IL6
##   <chr> <chr> <chr>    <dbl> <dbl> <dbl>      <dbl>      <dbl>
## 1 X1    A3    CAIR_86    430. 1934.  410.      222.      579.
## 2 X1    B3    CAIR_86    427. 1962.  421.      221.      587.
## 3 X2    C3    CAIR_47    328. 3246.  474.      171.      948.
## 4 X2    D3    CAIR_47    305. 3066.  466.      159.      898.
## 5 X3    E3    CAIR_67    396. 2088.  358.      205.      623.
## 6 X3    F3    CAIR_67    390. 1995.  357.      202.      596.
## 7 X4    G3    CAIR_96    743. 3594.  882.      386.     1043.
## 8 X4    H3    CAIR_96    746. 3521.  871.      387.     1023.
## 9 X5    A4    CAIR_130   315. 1836.  542.      164.      550.
## 10 X5   B4    CAIR_130   313. 1820.  543.      163.      546.
## # i 66 more rows
## # i 4 more variables: Predicted_TNFa <dbl>, Check_IL1b <chr>, Check_IL6 <chr>,
## #   Check_TNFa <chr>
```

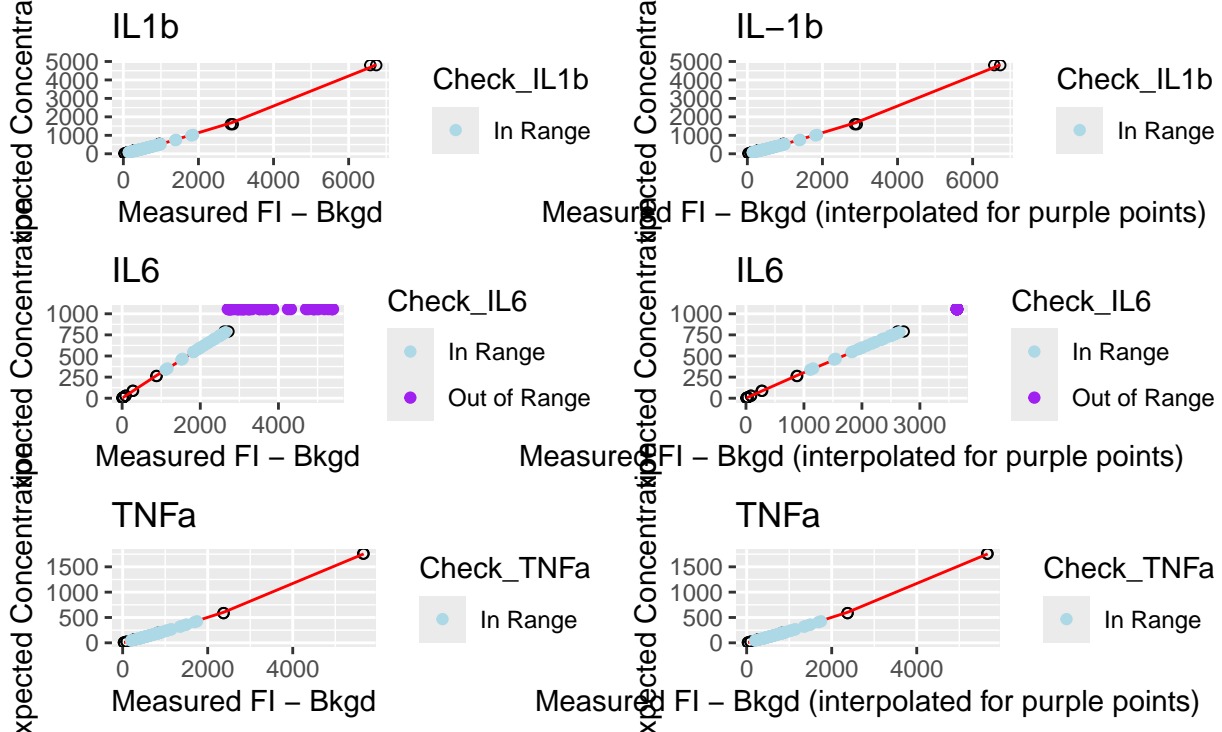
Interpolate values for samples flagged as “out of range”



Weighted Standard Curve (red) and data points. Black: standards.

Blue: sample values in range. Purple: values interpolated for out of range samples

Measured sample fluorescences shown (or interpolated) sample fluorescence s



```
## TableGrob (2 x 2) "arrange": 3 grobs
##   z   cells   name      grob
## 1 1 (2-2,1-1) arrange    gtable[arrange]
## 2 2 (2-2,2-2) arrange    gtable[arrange]
## 3 3 (1-1,1-2) arrange text[GRID.text.1901]
```

Calculate means and multiply by 30

```
# Calculate the mean of the duplicates based on the Description value and include the check value
mean_duplicates_2 <- reportable_data_2 %>%
  group_by(Description) %>%
  summarise(Mean_IL1b = (mean(report_IL1b))*30, #calculate mean and multiply by 30 to account for dilut
            Mean_IL6 = (mean(report_IL6))*30,
            Mean_TNFa = (mean(report_TNFa))*30,
            Check_IL1b = ifelse(any(Check_IL1b == "Out of Range"), "Out of Range", "In Range"),
            Check_IL6 = ifelse(any(Check_IL6 == "Out of Range"), "Out of Range", "In Range"),
            Check_TNFa = ifelse(any(Check_TNFa == "Out of Range"), "Out of Range", "In Range"))

# Display the mean of the duplicates
# mean_duplicates_2
```

Save results

```
# Save the unknown raw values to a CSV file
write.csv(unknown_data_2, here("unknown_raw_values_2.csv"), row.names = FALSE)

# Save the mean of the duplicates to a CSV file
readr::write_rds(mean_duplicates_2, here("luminex_results_2.rds"))
write.csv(mean_duplicates_2, here("luminex_results_2.csv"), row.names = FALSE)
```

Assay panel labelled 3

Load the Data

```
# Load the data from the Excel file
#####
# ASSAY 3
#####
file_path_3 <- here("R&D3plex_LX200_20240226_3.xlsx")
data_3 <- read_excel(file_path_3, sheet = "FI - Bkgd", range = "A63:F159")

## New names:
## * 'FI - Bkgd' -> 'FI - Bkgd...4'
## * 'FI - Bkgd' -> 'FI - Bkgd...5'
## * 'FI - Bkgd' -> 'FI - Bkgd...6'

# Rename columns for easier reference
colnames(data_3) <- c("Type", "Well", "Description", "IL1b", "IL6", "TNFa")
# Fill in missing Description values for the first 14 rows based on Type
data_3 <- data_3 %>%
  mutate(Description = ifelse(row_number() <= 14 & is.na(Description), Type, Description))
# Verify the changes
head(data_3, 20)
```

```
## # A tibble: 20 x 6
##   Type Well Description    IL1b    IL6    TNFa
##   <chr> <chr> <chr>      <dbl>  <dbl>  <dbl>
## 1 B     A1     B           8       5       7
## 2 B     B1     B           9       6       7
## 3 S1    C1     S1        7064.  2796.  5902.
## 4 S1    D1     S1        7124.  2896.  6097
## 5 S2    E1     S2        3022.   905.  2512.
## 6 S2    F1     S2        3268.   942.  2659
## 7 S3    G1     S3        1068.   302.   934
## 8 S3    H1     S3        1129.   306.   961
## 9 S4    A2     S4         312.    89.   297
## 10 S4   B2     S4         344.    99.   320.
## 11 S5    C2     S5         112.   32.5   106
## 12 S5    D2     S5         110.   33.5   110
```

```
## 13 S6      E2      S6          32.5      9.5      32
## 14 S6      F2      S6          36.5     10.5     36.5
## 15 C1      G2      IQC           7.5    114.     65.5
## 16 C1      H2      IQC           7      116      66
## 17 X1      A3      CAIR_255     602.    5256.    1412
## 18 X1      B3      CAIR_255     606     5287     1393
## 19 X2      C3      CAIR_143     403     4845     1006.
## 20 X2      D3      CAIR_143     410.    4805     990.
```

```
# Load the expected concentrations from the Excel file
```

```
exp_conc_3 <- read_excel(file_path_3, sheet = "Exp Conc", range = "D66:F77", col_names = FALSE)
```

```
## New names:
```

```
## * ' ' -> '...1'
```

```
## * ' ' -> '...2'
```

```
## * ' ' -> '...3'
```

```
# Set the correct column names
```

```
colnames(exp_conc_3) <- c("IL1b", "IL6", "TNFa")
```

```
# Extract the standard data and add expected concentrations
```

```
standard_data_3 <- data_3 %>%
```

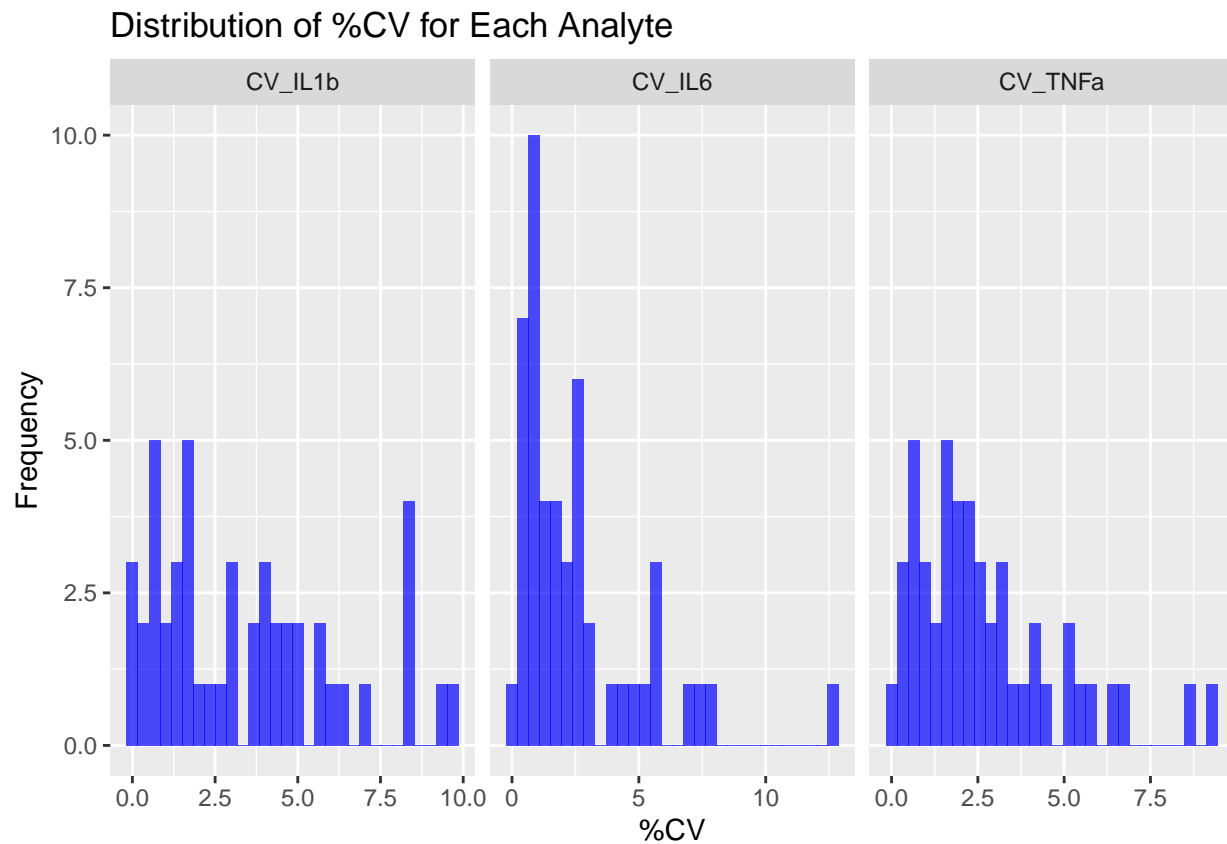
```
  filter(grepl("^S", Type)) %>%
```

```
  mutate(IL1b_exp = exp_conc_3$IL1b,
```

```
         IL6_exp = exp_conc_3$IL6,
```

```
         TNFa_exp = exp_conc_3$TNFa)
```


Calculate Mean and %CV for Replicates



```
## # A tibble: 48 x 4
##   Description CV_IL1b CV_IL6 CV_TNFa
##   <chr>      <dbl> <dbl> <dbl>
## 1 CAIR_264    9.70  7.71  8.74
## 2 UKO-003-t0  9.34  5.60  6.58
## 3 CAIR_265    8.48  5.54  4.22
## 4 CAIR_251    8.34  5.27  5.83
## 5 B           8.32 12.9   0
## 6 S6          8.20  7.07  9.29
## 7 S4          6.89  7.52  5.38
## 8 UKO-008-t0  6.47  0.866  1.90
## 9 UKO-006-t0  6.06  3.11  4.45
## 10 CAIR_262   5.64  4.53  5.17
## # i 38 more rows
```

Use weighting strategy to address poor fit at lower end of concentration range

```
##
## Call:
## lm(formula = IL1b_exp ~ poly(IL1b, 2), data = standard_data_3,
```

```

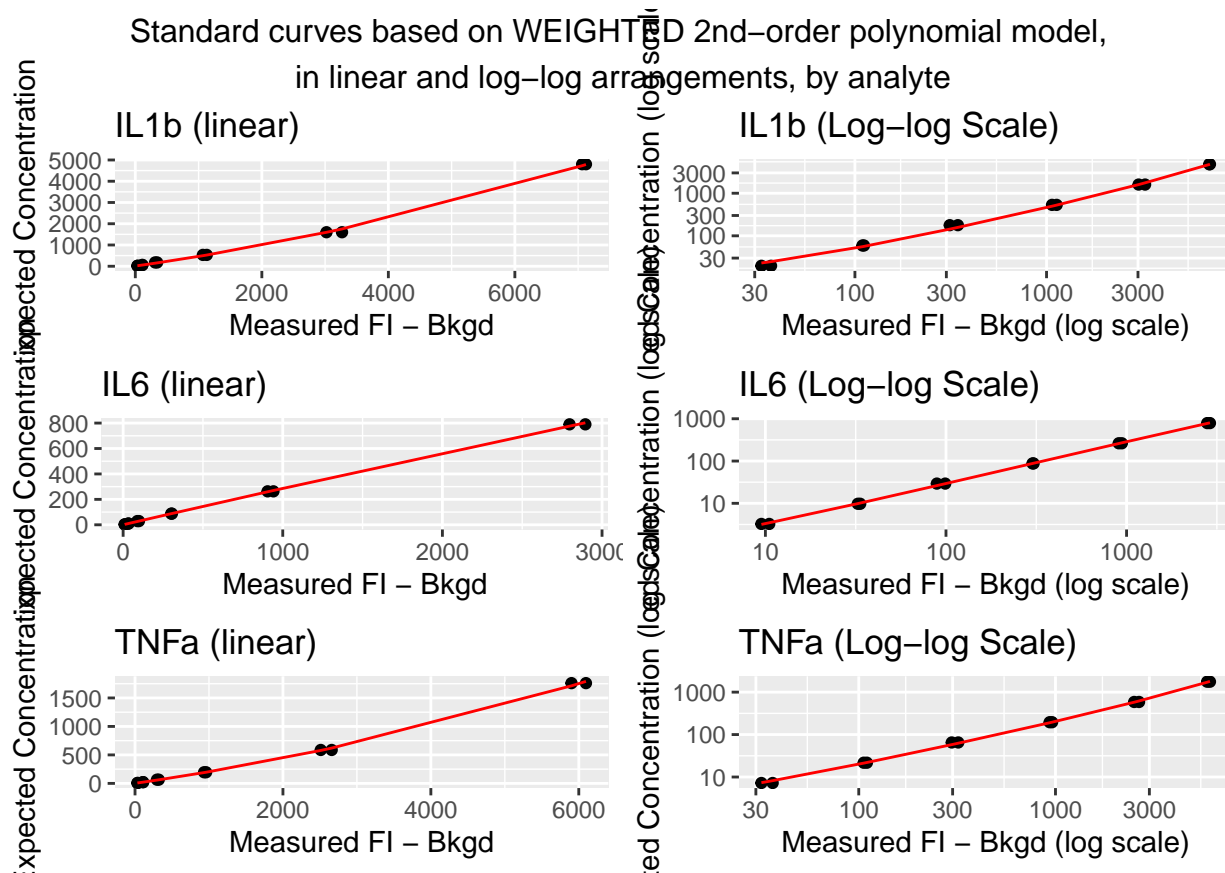
##      weights = weight_IL1b)
##
## Weighted Residuals:
##      Min      1Q   Median      3Q      Max
## -3.8525 -0.1240  0.3216  1.1358  2.5646
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1195.89      18.08  66.139 2.08e-13 ***
## poly(IL1b, 2)1  5832.05     105.18  55.449 1.02e-12 ***
## poly(IL1b, 2)2   548.85      61.37   8.943 8.99e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.811 on 9 degrees of freedom
## Multiple R-squared:  0.9978, Adjusted R-squared:  0.9973
## F-statistic: 2043 on 2 and 9 DF, p-value: 1.095e-12

##
## Call:
## lm(formula = IL6_exp ~ poly(IL6, 2), data = standard_data_3,
##     weights = weight_IL6)
##
## Weighted Residuals:
##      Min      1Q   Median      3Q      Max
## -0.43112 -0.11951 -0.00572  0.09157  0.55933
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    197.145      1.354  145.61 < 2e-16 ***
## poly(IL6, 2)1  967.741      8.199  118.03 1.14e-15 ***
## poly(IL6, 2)2  -10.381      4.475   -2.32  0.0455 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.334 on 9 degrees of freedom
## Multiple R-squared:  0.9995, Adjusted R-squared:  0.9994
## F-statistic: 9901 on 2 and 9 DF, p-value: 9.077e-16

##
## Call:
## lm(formula = TNFa_exp ~ poly(TNFa, 2), data = standard_data_3,
##     weights = weight_TNFa)
##
## Weighted Residuals:
##      Min      1Q   Median      3Q      Max
## -1.36202 -0.13582  0.03788  0.34338  1.11893
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    438.986      4.484   97.90 6.14e-15 ***
## poly(TNFa, 2)1 2143.107     26.192   81.82 3.08e-14 ***
## poly(TNFa, 2)2  195.728     14.892   13.14 3.53e-07 ***
## ---

```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7413 on 9 degrees of freedom
## Multiple R-squared:  0.999, Adjusted R-squared:  0.9988
## F-statistic: 4476 on 2 and 9 DF, p-value: 3.225e-14
```



Back-calculate standard curve observed values with weighted model

```
## # A tibble: 12 x 11
##   Type Well IL1b_exp Fitted_IL1b_weighted CV_IL1b_weighted IL6_exp
##   <chr> <chr>   <dbl>           <dbl>           <dbl>           <dbl>
## 1 S1    C1      4800           4730.           1.48            790
## 2 S1    D1      4800           4786.           0.303           790
## 3 S2    E1      1600           1597.           0.203           263.
## 4 S2    F1      1600           1754.           8.79            263.
## 5 S3    G1       533.           497.            7.41            87.8
## 6 S3    H1       533.           527.            1.24            87.8
## 7 S4    A2       178.           144.            23.8            29.3
## 8 S4    B2       178.           158.            12.7            29.3
## 9 S5    C2        59.3          56.5            4.91            9.75
## 10 S5   D2        59.3          55.6            6.52            9.75
## 11 S6    E2        19.8          23.0            14.3            3.25
## 12 S6    F2        19.8          24.7            20.1            3.25
## # i 5 more variables: Fitted_IL6_weighted <dbl>, CV_IL6_weighted <dbl>,
```

```
## # TNFa_exp <dbl>, Fitted_TNFa_weighted <dbl>, CV_TNFa_weighted <dbl>
```

```
## Warning in styling_latex_scale(out, table_info, "down"): Longtable cannot be
## resized.
```

Table 5: Less dramatic increase in CV values with decreasing expected concentration

Type	Well	IL1b_exp	CV_IL1b_weighted	IL6_exp	CV_IL6_weighted	TNFa_exp	CV_TNFa_weighted
S1	C1	4800.00	1.4835407	790.00	1.8021971	1760.00	2.7402220
S1	D1	4800.00	0.3027015	790.00	1.5106838	1760.00	1.7257678
S2	E1	1600.00	0.2028650	263.33	1.4755248	586.67	1.3610164
S2	F1	1600.00	8.7851577	263.33	2.3937514	586.67	5.3238646
S3	G1	533.33	7.4094686	87.78	0.2203607	195.56	2.9281260
S3	H1	533.33	1.2396420	87.78	1.0793279	195.56	0.1778729
S4	A2	177.78	23.8154176	29.26	11.5327379	65.19	12.3469274
S4	B2	177.78	12.7221893	29.26	0.4421453	65.19	4.0734134
S5	C2	59.26	4.9128990	9.75	1.0159030	21.73	2.2072282
S5	D2	59.26	6.5201736	9.75	3.8491737	21.73	1.3053418
S6	E2	19.75	14.2589655	3.25	2.4694356	7.24	1.6923933
S6	F2	19.75	20.0972816	3.25	6.1275922	7.24	11.7516842

Predict values for unknown samples

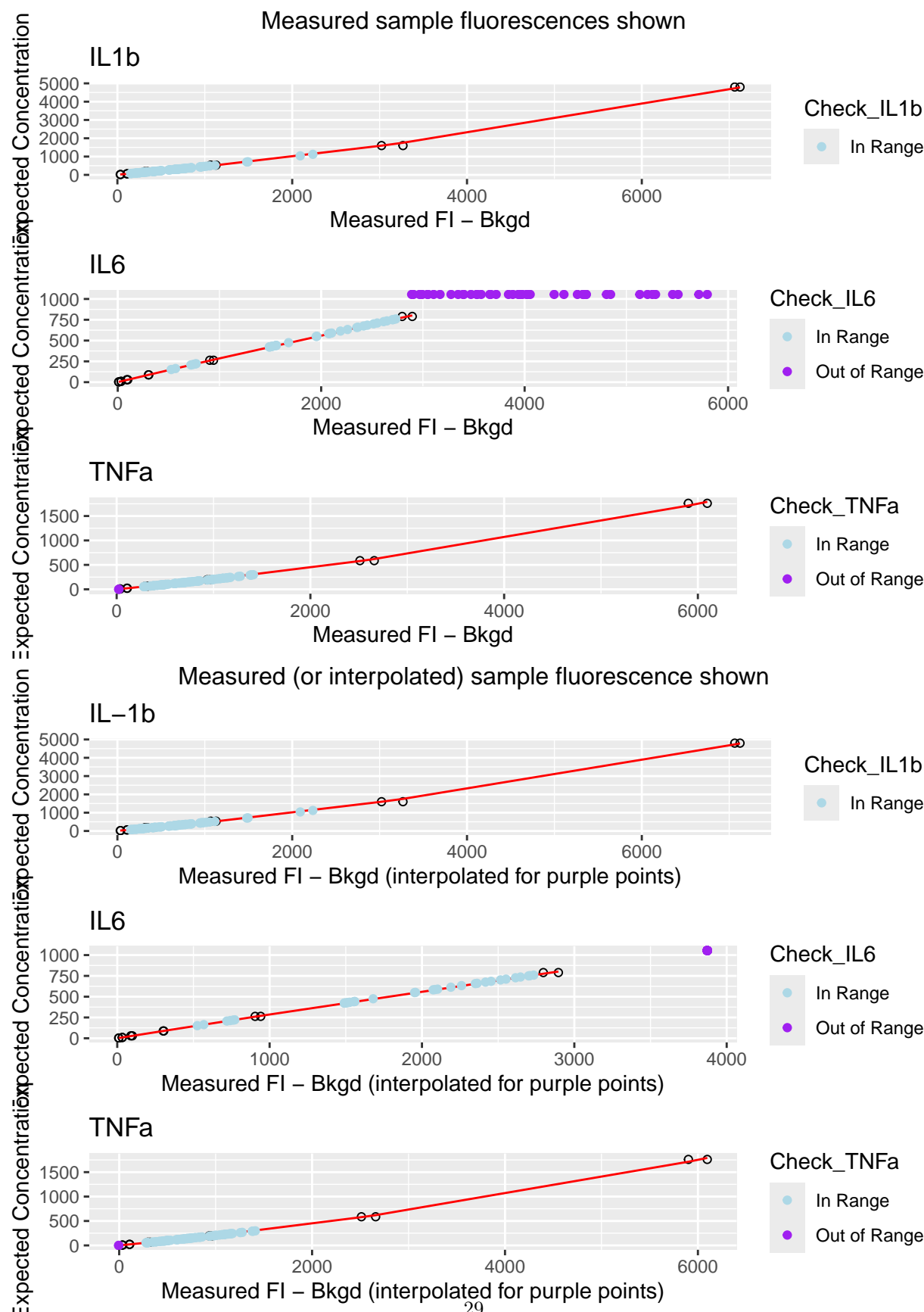
```
## # A tibble: 80 x 12
```

```
##   Type Well Description IL1b IL6 TNFa Predicted_IL1b Predicted_IL6
##   <chr> <chr> <chr>      <dbl> <dbl> <dbl>          <dbl>          <dbl>
## 1 X1    A3    CAIR_255    602.  5256.  1412          274.          1397.
## 2 X1    B3    CAIR_255    606  5287  1393          276.          1405.
## 3 X2    C3    CAIR_143    403  4845  1006.          184.          1297.
## 4 X2    D3    CAIR_143    410.  4805  990.           187.          1287.
## 5 X3    E3    CAIR_257    958.  4519  1142.          442.          1217.
## 6 X3    F3    CAIR_257    954  4604.  1153          441.          1238.
## 7 X4    G3    CAIR_212    437  4028.  602.           199.          1094.
## 8 X4    H3    CAIR_212    418  4054.  611           190.          1101.
## 9 X5    A4    CAIR_260   1056.  5508.  1378.          490.          1458.
## 10 X5   B4    CAIR_260   1116  5456.  1387          520.          1445.
```

```
## # i 70 more rows
```

```
## # i 4 more variables: Predicted_TNFa <dbl>, Check_IL1b <chr>, Check_IL6 <chr>,
## # Check_TNFa <chr>
```

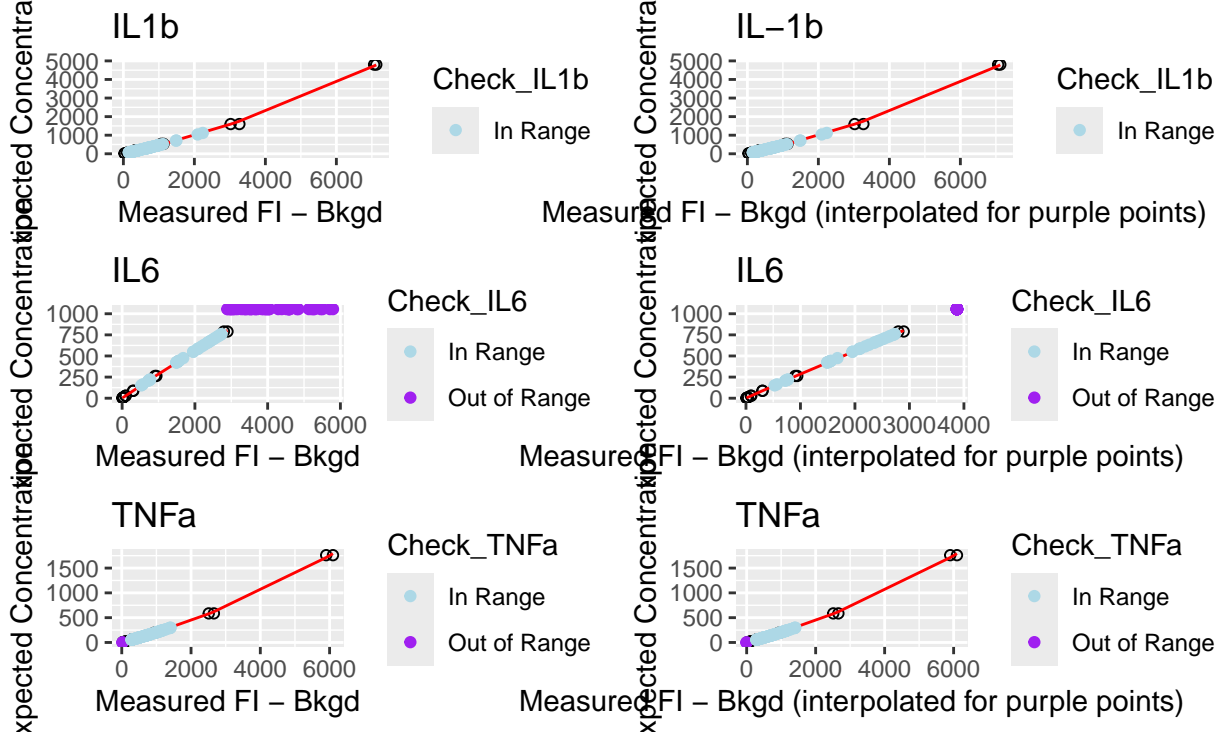
Interpolate values for samples flagged as “out of range”



Weighted Standard Curve (red) and data points. Black: standards.

Blue: sample values in range. Purple: values interpolated for out of range samples

Measured sample fluorescences shown (or interpolated) sample fluorescence s



```
## TableGrob (2 x 2) "arrange": 3 grobs
##   z   cells   name      grob
## 1 1 (2-2,1-1) arrange  gtable[arrange]
## 2 2 (2-2,2-2) arrange  gtable[arrange]
## 3 3 (1-1,1-2) arrange text[GRID.text.2623]
```

Calculate means and multiply by 30

```
# Calculate the mean of the duplicates based on the Description value and include the check value
mean_duplicates_3 <- reportable_data_3 %>%
  group_by(Description) %>%
  summarise(Mean_IL1b = (mean(report_IL1b))*30, #calculate mean and multiply by 30 to account for dilut
            Mean_IL6 = (mean(report_IL6))*30,
            Mean_TNFa = (mean(report_TNFa))*30,
            Check_IL1b = ifelse(any(Check_IL1b == "Out of Range"), "Out of Range", "In Range"),
            Check_IL6 = ifelse(any(Check_IL6 == "Out of Range"), "Out of Range", "In Range"),
            Check_TNFa = ifelse(any(Check_TNFa == "Out of Range"), "Out of Range", "In Range"))

# Display the mean of the duplicates
# mean_duplicates_3
```

Save results

```
# Save the unknown raw values to a CSV file
write.csv(unknown_data_3, here("unknown_raw_values_3.csv"), row.names = FALSE)

# Save the mean of the duplicates to a CSV file
readr::write_rds(mean_duplicates_3, here("luminex_results_3.rds"))
write.csv(mean_duplicates_3, here("luminex_results_3.csv"), row.names = FALSE)
```