

**Exploring the Working Range of Automated Standard Dilution  
Analysis of Nutrient Elements in Foods by Inductively Coupled  
Plasma Optical Emission Spectrometry**

Jake A. Carter\*, Patrick J. Gray, Todor I. Todorov

U.S. Food and Drug Administration, Center for Food Safety and Applied Nutrition, Office  
of Regulatory Science

5001 Campus Drive, College Park, MD, 20740, USA

\*Corresponding author: [jake.carter@fda.hhs.gov](mailto:jake.carter@fda.hhs.gov)

## **Supplementary Information**

**Table S1.** Results from the analysis of 10 different reference materials presented in tabular format.

**Table S2.** Results from the analysis of four different fortified foods spiked at three levels each presented in tabular format.

**R Code**

Table S1.

Element	Wavelength (nm)	Reference Material	Certificate	Figure of Merit Indicator	Conc. Mean (mg/kg)	Conc. Stdev. (mg/kg)	RSD (%)	Standard Concentration (mg/kg)	n	Z Score	Certificate Value (mg/kg)	Certificate Uncertainty (mg/kg)	Soln. Conc. Mean (mg/kg)	Soln. Conc. Stdev. (mg/kg)	Analytical Solution Detection Limit (ASDL)	Analytical Solution Quantitation Limit (ASQL)	Limit of Detection (LOD)	Limit of Quantitation (LOQ)
Ca	315.887	1566b	certified		784	8.29	1.06	9.8	3	-0.337	838	20	3.85	0.178	0.0286	0.0953	5.83	19.4
Ca	315.887	1566b	certified		775	14.7	1.9	9.8	3	-0.478	838	20	4.27	0.568	0.0286	0.0953	5.83	19.4
Ca	315.887	1568b	certified		106	1.34	1.26	9.8	3	-0.565	118.4	3.1	0.559	0.0208	0.0286	0.0953	5.83	19.4
Ca	315.887	1577c	certified		116	1.43	1.23	9.8	3	-0.801	131	10	0.557	0.0509	0.0286	0.0953	5.83	19.4
Ca	315.887	1577c	certified		115	1.63	1.42	9.8	3	-0.735	131	10	0.542	0.0154	0.0286	0.0953	5.83	19.4
Ca	315.887	1577c	certified		115	0.587	0.51	9.8	3	-0.823	131	10	0.537	0.00625	0.0286	0.0953	5.83	19.4
Ca	315.887	1845a	certified		2330	38.6	1.66	9.8	3	-0.152	2369	91	12.4	2.02	0.0286	0.0953	5.83	19.4
Ca	315.887	2976	reference		8650	2220	25.7	9.8	3	1.66	7600	300	44.7	12.6	0.0286	0.0953	5.83	19.4
Ca	315.887	3233	certified		53900	4520	8.39	9.8	3	3.43	36910	920	275	42.5	0.0286	0.0953	5.83	19.4
Ca	315.887	3233	certified		58800	10700	18.2	9.8	3	3.85	36910	920	352	93.3	0.0286	0.0953	5.83	19.4
Ca	315.887	3290	certified		11300	1090	9.65	9.8	3	0.534	11230	600	65.2	8.7	0.0286	0.0953	5.83	19.4
Ca	315.887	FDA cocoa	certified		1730	7.61	0.44	9.8	3	0.294	1677	47	8.32	0.341	0.0286	0.0953	5.83	19.4
Ca	315.887	dolt-4	information		625	12.8	2.05	9.8	3	NA	680	NA	3.35	0.317	0.0286	0.0953	5.83	19.4
Ca	315.887	dorm-4	certified		2300	92.8	4.03	9.8	3	0.322	2360	140	11.7	0.687	0.0286	0.0953	5.83	19.4
Cu	327.395	1566b	certified		68.1	0.631	0.927	0.98	3	-0.161	71.6	1.6	0.335	0.0138	0.00343	0.0114	0.699	2.33
Cu	327.395	1566b	certified		67.2	1.16	1.73	0.98	3	-0.335	71.6	1.6	0.37	0.0481	0.00343	0.0114	0.699	2.33
Cu	327.395	1568b	certified	Trace	1.62	0.149	9.2	0.98	3	-3.31	2.35	0.16	0.0086	0.00107	0.00343	0.0114	0.699	2.33
Cu	327.395	1577c	certified		268	5.77	2.15	0.98	3	0.156	275.2	4.6	1.29	0.135	0.00343	0.0114	0.699	2.33
Cu	327.395	1577c	certified		269	5.74	2.13	0.98	3	0.302	275.2	4.6	1.27	0.0422	0.00343	0.0114	0.699	2.33
Cu	327.395	1577c	certified		265	3.56	1.34	0.98	3	0.119	275.2	4.6	1.24	0.0111	0.00343	0.0114	0.699	2.33
Cu	327.395	1845a	certified	Trace	1.41	0.195	13.8	0.98	3	-6.51	2.4	0.11	0.0075	0.00134	0.00343	0.0114	0.699	2.33
Cu	327.395	2976	certified		3.21	0.172	5.36	0.98	3	-1.83	4.02	0.33	0.0165	0.00129	0.00343	0.0114	0.699	2.33
Cu	327.395	3233	certified		3.12	0.0587	1.88	0.98	3	-1.99	3.97	0.28	0.0159	0.00133	0.00343	0.0114	0.699	2.33
Cu	327.395	3233	certified		3.28	0.202	6.16	0.98	3	-1.71	3.97	0.28	0.0194	0.000426	0.00343	0.0114	0.699	2.33
Cu	327.395	3290	certified		15.6	0.218	1.4	0.98	3	-1.21	18.35	0.64	0.0912	0.0229	0.00343	0.0114	0.699	2.33
Cu	327.395	FDA cocoa	certified		42.7	0.38	0.89	0.98	3	0.0887	42.28	2.06	0.205	0.00827	0.00343	0.0114	0.699	2.33
Cu	327.395	dolt-4	certified		29.8	0.275	0.923	0.98	3	-0.0381	31.2	1.1	0.16	0.0133	0.00343	0.0114	0.699	2.33
Cu	327.395	dorm-4	certified		13.9	0.365	2.63	0.98	3	-0.643	15.7	0.46	0.0708	0.00315	0.00343	0.0114	0.699	2.33

Fe	238.204	1566b	certified	196	2.76	1.41	0.98	3	-0.159	205.8	6.8	0.962	0.0418	0.0055	0.0183	1.12	3.74
Fe	238.204	1566b	certified	198	4.78	2.41	0.98	3	-0.102	205.8	6.8	1.09	0.139	0.0055	0.0183	1.12	3.74
Fe	238.204	1568b	certified	6.97	1.04	14.9	0.98	3	-0.0518	7.42	0.44	0.0368	0.00533	0.0055	0.0183	1.12	3.74
Fe	238.204	1577c	certified	202	7.98	3.95	0.98	3	0.576	197.94	0.65	0.969	0.0625	0.0055	0.0183	1.12	3.74
Fe	238.204	1577c	certified	199	5.22	2.62	0.98	3	0.57	197.94	0.65	0.937	0.0226	0.0055	0.0183	1.12	3.74
Fe	238.204	1577c	certified	196	1.88	0.959	0.98	3	0.381	197.94	0.65	0.919	0.00633	0.0055	0.0183	1.12	3.74
Fe	238.204	1845a	certified	81.4	2.33	2.86	0.98	3	-0.193	83	2.3	0.433	0.0668	0.0055	0.0183	1.12	3.74
Fe	238.204	2976	certified	167	1.46	0.874	0.98	3	0.314	171	4.9	0.862	0.027	0.0055	0.0183	1.12	3.74
Fe	238.204	3233	certified	761	20.5	2.69	0.98	3	0.364	766	36	3.87	0.325	0.0055	0.0183	1.12	3.74
Fe	238.204	3233	certified	771	21.1	2.74	0.98	3	0.275	766	36	4.58	0.481	0.0055	0.0183	1.12	3.74
Fe	238.204	3290	certified	309	8.25	2.67	0.98	3	-1.21	364	15	1.81	0.481	0.0055	0.0183	1.12	3.74
Fe	238.204	FDA cocoa	certified	129	1.54	1.19	0.98	3	0.00559	129.2	2.6	0.622	0.0205	0.0055	0.0183	1.12	3.74
Fe	238.204	dolt-4	certified	2190	68.4	3.12	0.98	3	1.97	1833	75	11.8	1.19	0.0055	0.0183	1.12	3.74
Fe	238.204	dorm-4	certified	331	5.61	1.69	0.98	3	0.213	343	20	1.69	0.0431	0.0055	0.0183	1.12	3.74
K	766.491	1566b	certified	5740	76.8	1.34	49	3	-0.978	6520	90	28.2	1.35	0.517	1.72	105	351
K	766.491	1566b	certified	5740	46.3	0.807	49	3	-1.01	6520	90	31.6	4.44	0.517	1.72	105	351
K	766.491	1568b	certified	915	17	1.86	49	3	-3.21	1282	11	4.84	0.202	0.517	1.72	105	351
K	766.491	1577c	certified	9400	47.7	0.507	49	3	-0.427	10230	640	45.3	4.08	0.517	1.72	105	351
K	766.491	1577c	certified	9400	23	0.245	49	3	-0.28	10230	640	44.4	1.12	0.517	1.72	105	351
K	766.491	1577c	certified	9370	93.6	0.999	49	3	-0.38	10230	640	43.9	0.24	0.517	1.72	105	351
K	766.491	1845a	certified	3640	50.3	1.38	49	3	-1.13	4110	400	19.4	3.25	0.517	1.72	105	351
K	766.491	2976	reference	9280	31.8	0.343	49	3	0.0946	9700	500	47.8	1.2	0.517	1.72	105	351
K	766.491	3233	certified	2530	50.5	2	49	3	-1.51	3060	140	12.9	1.28	0.517	1.72	105	351
K	766.491	3233	certified	2600	29.5	1.13	49	3	-1.45	3060	140	15.4	1.42	0.517	1.72	105	351
K	766.491	3290	certified	6560	21.6	0.329	49	3	-1.11	7660	250	38.3	8.94	0.517	1.72	105	351
K	766.491	FDA cocoa	certified	17800	114	0.64	49	3	-0.121	17981	431	85.5	3.26	0.517	1.72	105	351
K	766.491	dolt-4	information	8290	62.6	0.755	49	3	NA	9800	NA	44.5	3.63	0.517	1.72	105	351
K	766.491	dorm-4	certified	11700	102	0.872	49	3	-2.33	15500	1000	59.5	1.61	0.517	1.72	105	351
Mg	279.8	1566b	certified	1070	3.42	0.32	9.8	3	0.195	1085	23	5.25	0.196	0.0514	0.171	10.5	34.9
Mg	279.8	1566b	certified	1050	24.9	2.37	9.8	3	0.00771	1085	23	5.78	0.722	0.0514	0.171	10.5	34.9
Mg	279.8	1568b	certified	476	17.1	3.59	9.8	3	-1.07	559	10	2.52	0.152	0.0514	0.171	10.5	34.9
Mg	279.8	1577c	certified	592	3.23	0.546	9.8	3	-0.057	620	42	2.85	0.231	0.0514	0.171	10.5	34.9

Mg	279.8	1577c	certified		590	8.63	1.46	9.8	3	0.0495	620	42	2.78	0.0623	0.0514	0.171	10.5	34.9
Mg	279.8	1577c	certified		588	3.87	0.658	9.8	3	-0.0474	620	42	2.75	0.00934	0.0514	0.171	10.5	34.9
Mg	279.8	1845a	certified		399	13.3	3.33	9.8	3	-0.168	406	12	2.12	0.322	0.0514	0.171	10.5	34.9
Mg	279.8	2976	reference		4630	51	1.1	9.8	3	-0.8	5300	500	23.8	0.831	0.0514	0.171	10.5	34.9
Mg	279.8	3233	certified		1040	9.98	0.96	9.8	3	-0.0388	1093	37	5.29	0.432	0.0514	0.171	10.5	34.9
Mg	279.8	3233	certified		1050	4.69	0.447	9.8	3	-0.19	1093	37	6.22	0.496	0.0514	0.171	10.5	34.9
Mg	279.8	3290	certified		921	16.8	1.82	9.8	3	-0.14	981	84	5.37	1.17	0.0514	0.171	10.5	34.9
Mg	279.8	FDA cocoa	certified		6110	94	1.54	9.8	3	0.333	5908	123	29.4	0.941	0.0514	0.171	10.5	34.9
Mg	279.8	dolt-4	information		1340	33.7	2.51	9.8	3	NA	1500	NA	7.2	0.591	0.0514	0.171	10.5	34.9
Mg	279.8	dorm-4	certified		850	28.6	3.36	9.8	3	-0.0858	910	80	4.33	0.22	0.0514	0.171	10.5	34.9
Mn	257.61	1566b	certified		17.6	0.169	0.96	0.98	3	-0.192	18.5	0.2	0.0862	0.00414	0.00208	0.00694	0.425	1.42
Mn	257.61	1566b	certified		17.5	0.24	1.37	0.98	3	-0.27	18.5	0.2	0.0961	0.0129	0.00208	0.00694	0.425	1.42
Mn	257.61	1568b	certified		18.2	0.486	2.67	0.98	3	0.018	19.2	1.8	0.096	0.00545	0.00208	0.00694	0.425	1.42
Mn	257.61	1577c	certified		9.56	0.0731	0.765	0.98	3	-0.487	10.46	0.47	0.046	0.00376	0.00208	0.00694	0.425	1.42
Mn	257.61	1577c	certified		9.51	0.16	1.68	0.98	3	-0.396	10.46	0.47	0.0449	0.000955	0.00208	0.00694	0.425	1.42
Mn	257.61	1577c	certified		9.35	0.0501	0.536	0.98	3	-0.641	10.46	0.47	0.0438	5.00E-04	0.00208	0.00694	0.425	1.42
Mn	257.61	1845a	certified	Trace	0.604	0.152	25.2	0.98	3	-6.69	1.15	0.11	0.00326	0.00116	0.00208	0.00694	0.425	1.42
Mn	257.61	2976	reference		35.4	0.592	1.67	0.98	3	1.15	33	2	0.182	0.00742	0.00208	0.00694	0.425	1.42
Mn	257.61	3233	certified		30.6	0.246	0.804	0.98	3	-0.339	33.1	1.1	0.156	0.0134	0.00208	0.00694	0.425	1.42
Mn	257.61	3233	certified		30.7	0.204	0.664	0.98	3	-0.527	33.1	1.1	0.182	0.0152	0.00208	0.00694	0.425	1.42
Mn	257.61	3290	certified		69.2	1.31	1.89	0.98	3	-0.178	74	6.8	0.404	0.0947	0.00208	0.00694	0.425	1.42
Mn	257.61	FDA cocoa	certified		53.1	0.231	0.435	0.98	3	0.264	51.68	1.55	0.256	0.0105	0.00208	0.00694	0.425	1.42
Mn	257.61	dolt-4	NA		8.73	0.347	3.97	0.98	3	NA	NA	NA	0.0468	0.00497	0.00208	0.00694	0.425	1.42
Mn	257.61	dorm-4	certified		2.01	0.155	7.71	0.98	3	-4.16	3.17	0.26	0.0103	0.000958	0.00208	0.00694	0.425	1.42
Na	588.995	1566b	certified		3440	44.4	1.29	49	3	0.719	3297	53	16.9	0.707	0.538	1.79	110	366
Na	588.995	1566b	certified		3400	96.5	2.84	49	3	0.585	3297	53	18.7	2.24	0.538	1.79	110	366
Na	588.995	1568b	certified	Trace	329	20.2	6.14	49	3	9.81	6.74	0.19	1.74	0.0505	0.538	1.79	110	366
Na	588.995	1577c	certified		2290	37.8	1.65	49	3	1.46	2033	64	11	0.795	0.538	1.79	110	366
Na	588.995	1577c	certified		2310	20.4	0.883	49	3	1.68	2033	64	10.9	0.232	0.538	1.79	110	366
Na	588.995	1577c	certified		2270	11.6	0.511	49	3	1.44	2033	64	10.6	0.036	0.538	1.79	110	366
Na	588.995	1845a	certified		4810	108	2.25	49	3	0.492	4576	93	25.6	3.79	0.538	1.79	110	366
Na	588.995	2976	reference		34000	246	0.724	49	3	0.246	35000	1000	175	3.06	0.538	1.79	110	366

Na	588.995	3233	certified		6800	75.2	1.11	49	3	0.387	6830	120	34.5	2.81	0.538	1.79	110	366
Na	588.995	3233	certified		6750	20.7	0.307	49	3	0.101	6830	120	40	3.31	0.538	1.79	110	366
Na	588.995	3290	certified		5030	120	2.39	49	3	-0.405	5500	230	29.3	6.22	0.538	1.79	110	366
Na	588.995	FDA cocoa	certified	Trace	337	20	5.93	49	3	9.4	19.89	2.02	1.62	0.0396	0.538	1.79	110	366
Na	588.995	dolt-4	information		6690	32.1	0.48	49	3	NA	6800	NA	35.8	2.49	0.538	1.79	110	366
Na	588.995	dorm-4	reference		13700	178	1.3	49	3	NA	14000	2400	69.9	2.13	0.538	1.79	110	366
P	213.618	1566b	NA		6740	117	1.74	9.8	3	NA	NA	NA	33.1	1.27	0.0287	0.0957	5.86	19.5
P	213.618	1566b	NA		6610	74.9	1.13	9.8	3	NA	NA	NA	36.4	4.98	0.0287	0.0957	5.86	19.5
P	213.618	1568b	certified		1380	30.6	2.22	9.8	3	-0.464	1530	40	7.29	0.375	0.0287	0.0957	5.86	19.5
P	213.618	1577c	reference		9620	142	1.48	9.8	3	NA	11750	270	46.3	3.48	0.0287	0.0957	5.86	19.5
P	213.618	1577c	reference		9790	359	3.67	9.8	3	NA	11750	270	46.2	0.833	0.0287	0.0957	5.86	19.5
P	213.618	1577c	reference		9650	133	1.38	9.8	3	NA	11750	270	45.2	0.475	0.0287	0.0957	5.86	19.5
P	213.618	1845a	certified		7770	199	2.56	9.8	3	-1.04	8640	610	41.3	5.9	0.0287	0.0957	5.86	19.5
P	213.618	2976	NA		6990	80	1.14	9.8	3	NA	NA	NA	36	1.24	0.0287	0.0957	5.86	19.5
P	213.618	3233	certified		2430	31	1.28	9.8	3	-0.208	2592	68	12.3	1	0.0287	0.0957	5.86	19.5
P	213.618	3233	certified		2440	24.6	1.01	9.8	3	-0.379	2592	68	14.5	1.35	0.0287	0.0957	5.86	19.5
P	213.618	3290	certified		8240	595	7.22	9.8	3	-0.744	9320	510	47.7	7.58	0.0287	0.0957	5.86	19.5
P	213.618	FDA cocoa	certified		7950	213	2.68	9.8	3	-0.425	8300	392	38.3	0.717	0.0287	0.0957	5.86	19.5
P	213.618	dolt-4	NA		9950	98.6	0.991	9.8	3	NA	NA	NA	53.4	4.34	0.0287	0.0957	5.86	19.5
P	213.618	dorm-4	information		7080	97	1.37	9.8	3	NA	8000	NA	36.1	1.16	0.0287	0.0957	5.86	19.5
S	181.972	1566b	certified		6150	32	0.52	9.8	3	-0.833	6890	140	30.2	1.3	0.0665	0.222	13.6	45.2
S	181.972	1566b	certified		6030	141	2.34	9.8	3	-1.08	6890	140	33.2	4.33	0.0665	0.222	13.6	45.2
S	181.972	1568b	certified		994	24.8	2.49	9.8	3	-1.39	1200	10	5.26	0.305	0.0665	0.222	13.6	45.2
S	181.972	1577c	certified		6720	94	1.4	9.8	3	-0.686	7490	340	32.4	3.06	0.0665	0.222	13.6	45.2
S	181.972	1577c	certified		6840	86.1	1.26	9.8	3	-0.349	7490	340	32.3	0.489	0.0665	0.222	13.6	45.2
S	181.972	1577c	certified		6770	55.8	0.824	9.8	3	-0.518	7490	340	31.7	0.26	0.0665	0.222	13.6	45.2
S	181.972	1845a	NA		5420	119	2.2	9.8	3	NA	NA	NA	28.8	4.8	0.0665	0.222	13.6	45.2
S	181.972	2976	NA		16100	102	0.634	9.8	3	NA	NA	NA	83.1	2.48	0.0665	0.222	13.6	45.2
S	181.972	3233	NA		873	12.8	1.47	9.8	3	NA	NA	NA	4.43	0.322	0.0665	0.222	13.6	45.2
S	181.972	3233	NA		875	4.51	0.515	9.8	3	NA	NA	NA	5.19	0.429	0.0665	0.222	13.6	45.2
S	181.972	3290	NA		4840	206	4.26	9.8	3	NA	NA	NA	28.1	5.36	0.0665	0.222	13.6	45.2
S	181.972	FDA cocoa	NA		2870	12.6	0.439	9.8	3	NA	NA	NA	13.8	0.588	0.0665	0.222	13.6	45.2

S	181.972	dolt-4	NA	12600	124	0.984	9.8	3	NA	NA	NA	67.5	5.69	0.0665	0.222	13.6	45.2
S	181.972	dorm-4	NA	8280	90.1	1.09	9.8	3	NA	NA	NA	42.2	0.592	0.0665	0.222	13.6	45.2
Zn	202.548	1566b	certified	2220	61.9	2.79	0.98	3	3.78	1424	46	10.9	0.694	0.00193	0.00642	0.393	1.31
Zn	202.548	1566b	certified	2240	163	7.28	0.98	3	3.8	1424	46	12.4	2.74	0.00193	0.00642	0.393	1.31
Zn	202.548	1568b	certified	17	0.444	2.61	0.98	3	-0.765	19.42	0.26	0.09	0.00501	0.00193	0.00642	0.393	1.31
Zn	202.548	1577c	certified	188	0.586	0.312	0.98	3	0.747	181.1	1	0.905	0.0766	0.00193	0.00642	0.393	1.31
Zn	202.548	1577c	certified	188	2.68	1.43	0.98	3	0.884	181.1	1	0.887	0.0188	0.00193	0.00642	0.393	1.31
Zn	202.548	1577c	certified	187	0.634	0.339	0.98	3	0.76	181.1	1	0.875	0.00407	0.00193	0.00642	0.393	1.31
Zn	202.548	1845a	certified	55.5	0.799	1.44	0.98	3	-0.0794	55.9	2.3	0.295	0.0475	0.00193	0.00642	0.393	1.31
Zn	202.548	2976	certified	138	1.58	1.14	0.98	3	0.608	137	13	0.713	0.0241	0.00193	0.00642	0.393	1.31
Zn	202.548	3233	certified	607	2.06	0.339	0.98	3	0.102	628	16	3.08	0.263	0.00193	0.00642	0.393	1.31
Zn	202.548	3233	certified	618	10.6	1.72	0.98	3	0.0662	628	16	3.67	0.349	0.00193	0.00642	0.393	1.31
Zn	202.548	3290	reference	214	17.5	8.18	0.98	3	-0.04	226	33	1.24	0.201	0.00193	0.00642	0.393	1.31
Zn	202.548	FDA cocoa	certified	79.7	0.405	0.508	0.98	3	0.178	78.3	3.2	0.384	0.017	0.00193	0.00642	0.393	1.31
Zn	202.548	dolt-4	certified	118	0.927	0.786	0.98	3	0.578	116	6	0.633	0.0502	0.00193	0.00642	0.393	1.31
Zn	202.548	dorm-4	certified	49.4	0.814	1.65	0.98	3	0.134	51.6	2.8	0.251	0.00876	0.00193	0.00642	0.393	1.31

Table S2.

Element	Wavelength (nm)	Sample	Fortified Analytical Portion (FAP) Level	Figure of Merit Indicator	Recovery Mean (%)	Recovery Stdev. (%)	RSD (%)	Standard Concentration (mg/kg)	n	Native Conc. Mean (mg/kg)	Native Conc. Stdev. (mg/kg)	Soln. Conc. Mean (mg/kg)	Soln. Conc. Stdev. (mg/kg)	Spike Conc. (mg/kg)	Analytical Solution Detection Limit (ASDL)	Analytical Solution Quantitation Limit (ASQL)	Limit of Detection (LOD)	Limit of Quantitation (LOQ)
Ca	315.887	Breaded Chicken	Level 1		99.6	17	17.1	9.8	3	96.7	0.618	0.918	0.17	71.3	0.0286	0.0953	5.83	19.4
Ca	315.887	Breaded Chicken	Level 2		98.8	8.07	8.17	9.8	3	96.7	0.618	1.16	0.0445	134	0.0286	0.0953	5.83	19.4
Ca	315.887	Breaded Chicken	Level 3		96.9	7.48	7.72	9.8	3	96.7	0.618	2.24	0.104	336	0.0286	0.0953	5.83	19.4
Ca	315.887	Corn Flakes	Level 1		53.5	14.3	26.7	9.8	3	20.6	11.8	0.135	0.0133	12.8	0.0286	0.0953	5.83	19.4
Ca	315.887	Corn Flakes	Level 2		74.6	4.22	5.66	9.8	3	20.6	11.8	0.209	0.015	25.7	0.0286	0.0953	5.83	19.4
Ca	315.887	Corn Flakes	Level 3		87.6	0.817	0.933	9.8	3	20.6	11.8	0.384	0.00768	63.6	0.0286	0.0953	5.83	19.4
Ca	315.887	Italian Dressing	Level 1		98.1	3.48	3.55	9.8	3	98.3	1.21	0.744	0.0147	53.3	0.0286	0.0953	5.83	19.4
Ca	315.887	Italian Dressing	Level 2		107	9.44	8.82	9.8	3	98.3	1.21	1.1	0.1	117	0.0286	0.0953	5.83	19.4
Ca	315.887	Italian Dressing	Level 3		94.2	0.962	1.02	9.8	3	98.3	1.21	1.84	0.0365	273	0.0286	0.0953	5.83	19.4
Ca	315.887	Vegetable Oil	Level 1	< LOD	-137	21.5	-15.7	9.8	3	-15.4	1.22	-0.0234	0.0023	6.69	0.0286	0.0953	5.83	19.4
Ca	315.887	Vegetable Oil	Level 2	< LOD	-16.5	19.5	-118	9.8	3	-15.4	1.22	-0.00603	0.00715	14.1	0.0286	0.0953	5.83	19.4
Ca	315.887	Vegetable Oil	Level 3	Trace	57	9.09	15.9	9.8	3	-15.4	1.22	0.0568	0.00947	40.2	0.0286	0.0953	5.83	19.4
Cu	327.395	Breaded Chicken	Level 1	< LOD	80.9	55.8	69	0.98	3	-0.0152	0.0546	0.00188	0.00138	0.377	0.00343	0.0114	0.699	2.33
Cu	327.395	Breaded Chicken	Level 2	< LOD	91	7.43	8.16	0.98	3	-0.0152	0.0546	0.00328	0.000238	0.711	0.00343	0.0114	0.699	2.33
Cu	327.395	Breaded Chicken	Level 3	Trace	99.9	8.57	8.58	0.98	3	-0.0152	0.0546	0.0094	0.000338	1.78	0.00343	0.0114	0.699	2.33
Cu	327.395	Corn Flakes	Level 1	Trace	86.3	7.08	8.2	0.98	3	-0.235	0.0868	0.00882	0.00103	2.06	0.00343	0.0114	0.699	2.33
Cu	327.395	Corn Flakes	Level 2		91.6	2.07	2.26	0.98	3	-0.235	0.0868	0.02	0.00133	4.16	0.00343	0.0114	0.699	2.33
Cu	327.395	Corn Flakes	Level 3		97.5	1.32	1.35	0.98	3	-0.235	0.0868	0.0505	0.00136	10.3	0.00343	0.0114	0.699	2.33
Cu	327.395	Italian Dressing	Level 1	Trace	78.6	6.98	8.88	0.98	3	-0.412	0.126	0.00565	0.000449	1.46	0.00343	0.0114	0.699	2.33
Cu	327.395	Italian Dressing	Level 2		81.7	4.9	6	0.98	3	-0.412	0.126	0.0128	0.000288	3.21	0.00343	0.0114	0.699	2.33
Cu	327.395	Italian Dressing	Level 3		93.9	2.69	2.86	0.98	3	-0.412	0.126	0.0362	0.00121	7.48	0.00343	0.0114	0.699	2.33
Cu	327.395	Vegetable Oil	Level 1	< LOD	35.7	5.33	14.9	0.98	3	-0.962	0.151	0.00183	0.000339	1.98	0.00343	0.0114	0.699	2.33
Cu	327.395	Vegetable Oil	Level 2	Trace	78.5	5.73	7.3	0.98	3	-0.962	0.151	0.00848	0.000685	4.16	0.00343	0.0114	0.699	2.33
Cu	327.395	Vegetable Oil	Level 3		94.1	3.3	3.51	0.98	3	-0.962	0.151	0.0278	0.0014	11.9	0.00343	0.0114	0.699	2.33
Fe	238.204	Breaded Chicken	Level 1		85.7	24.4	28.5	0.98	3	10.2	0.238	0.0781	0.0129	4.83	0.0055	0.0183	1.12	3.74
Fe	238.204	Breaded Chicken	Level 2		95.2	9.91	10.4	0.98	3	10.2	0.238	0.0955	0.00407	9.1	0.0055	0.0183	1.12	3.74
Fe	238.204	Breaded Chicken	Level 3		94	6.28	6.68	0.98	3	10.2	0.238	0.167	0.00746	22.8	0.0055	0.0183	1.12	3.74
Fe	238.204	Corn Flakes	Level 1		107	3.73	3.49	0.98	3	370	7.36	2.8	0.136	185	0.0055	0.0183	1.12	3.74

Fe	238.204	Corn Flakes	Level 2		104	2.33	2.24	0.98	3	370	7.36	3.98	0.246	372	0.0055	0.0183	1.12	3.74
Fe	238.204	Corn Flakes	Level 3		108	4.55	4.21	0.98	3	370	7.36	6.88	0.307	922	0.0055	0.0183	1.12	3.74
Fe	238.204	Italian Dressing	Level 1	Trace	113	20.8	18.4	0.98	3	2.12	0.125	0.0166	0.00116	1.11	0.0055	0.0183	1.12	3.74
Fe	238.204	Italian Dressing	Level 2		90.9	5.89	6.48	0.98	3	2.12	0.125	0.0212	0.000276	2.44	0.0055	0.0183	1.12	3.74
Fe	238.204	Italian Dressing	Level 3		104	20.8	20	0.98	3	2.12	0.125	0.0416	0.00671	5.68	0.0055	0.0183	1.12	3.74
Fe	238.204	Vegetable Oil	Level 1	< LOD	136	16.8	12.4	0.98	3	0.357	0.092	0.00487	0.000578	1.39	0.0055	0.0183	1.12	3.74
Fe	238.204	Vegetable Oil	Level 2	Trace	110	6.65	6.05	0.98	3	0.357	0.092	0.00834	0.000525	2.92	0.0055	0.0183	1.12	3.74
Fe	238.204	Vegetable Oil	Level 3		109	1.73	1.59	0.98	3	0.357	0.092	0.0225	0.000622	8.36	0.0055	0.0183	1.12	3.74
K	766.491	Breaded Chicken	Level 1		99.5	17.3	17.4	49	3	2840	45	25.5	4.51	1830	0.517	1.72	105	351
K	766.491	Breaded Chicken	Level 2		97.8	6.61	6.76	49	3	2840	45	31.7	1.46	3480	0.517	1.72	105	351
K	766.491	Breaded Chicken	Level 3		97	7.49	7.72	49	3	2840	45	59.8	2.78	8700	0.517	1.72	105	351
K	766.491	Corn Flakes	Level 1		96	7.75	8.07	49	3	698	24.6	5.71	0.373	478	0.517	1.72	105	351
K	766.491	Corn Flakes	Level 2		94	1.52	1.62	49	3	698	24.6	8.43	0.497	962	0.517	1.72	105	351
K	766.491	Corn Flakes	Level 3		96.1	1.65	1.72	49	3	698	24.6	15	0.417	2380	0.517	1.72	105	351
K	766.491	Italian Dressing	Level 1		92	5.24	5.7	49	3	375	11.6	3.01	0.0882	255	0.517	1.72	105	351
K	766.491	Italian Dressing	Level 2		84.4	4.7	5.57	49	3	375	11.6	4.15	0.0836	560	0.517	1.72	105	351
K	766.491	Italian Dressing	Level 3		92.4	1.94	2.1	49	3	375	11.6	8.17	0.116	1310	0.517	1.72	105	351
K	766.491	Vegetable Oil	Level 1	Trace	58.2	6.13	10.5	49	3	-130	24.5	0.856	0.162	566	0.517	1.72	105	351
K	766.491	Vegetable Oil	Level 2		73.9	2.85	3.86	49	3	-130	24.5	2.27	0.0876	1180	0.517	1.72	105	351
K	766.491	Vegetable Oil	Level 3		83.6	2.74	3.28	49	3	-130	24.5	7.04	0.354	3400	0.517	1.72	105	351
Mg	279.8	Breaded Chicken	Level 1		102	19.6	19.2	9.8	3	255	5.49	2.27	0.382	158	0.0514	0.171	10.5	34.9
Mg	279.8	Breaded Chicken	Level 2		103	7.77	7.54	9.8	3	255	5.49	2.85	0.118	298	0.0514	0.171	10.5	34.9
Mg	279.8	Breaded Chicken	Level 3		101	7.19	7.12	9.8	3	255	5.49	5.34	0.24	746	0.0514	0.171	10.5	34.9
Mg	279.8	Corn Flakes	Level 1		101	7	6.93	9.8	3	89.6	2.42	0.694	0.041	50.2	0.0514	0.171	10.5	34.9
Mg	279.8	Corn Flakes	Level 2		97.7	1.18	1.21	9.8	3	89.6	2.42	0.99	0.0538	101	0.0514	0.171	10.5	34.9
Mg	279.8	Corn Flakes	Level 3		99	0.635	0.641	9.8	3	89.6	2.42	1.7	0.0279	250	0.0514	0.171	10.5	34.9
Mg	279.8	Italian Dressing	Level 1		104	4.19	4.03	9.8	3	32.8	1.23	0.257	0.00564	18.5	0.0514	0.171	10.5	34.9
Mg	279.8	Italian Dressing	Level 2		97.4	5.98	6.14	9.8	3	32.8	1.23	0.354	0.00841	40.7	0.0514	0.171	10.5	34.9
Mg	279.8	Italian Dressing	Level 3		110	12.4	11.3	9.8	3	32.8	1.23	0.709	0.0709	94.9	0.0514	0.171	10.5	34.9
Mg	279.8	Vegetable Oil	Level 1	Trace	107	4.42	4.13	9.8	3	0.0928	3.05	0.0831	0.00944	30.1	0.0514	0.171	10.5	34.9
Mg	279.8	Vegetable Oil	Level 2	Trace	104	7.6	7.31	9.8	3	0.0928	3.05	0.17	0.0115	63.3	0.0514	0.171	10.5	34.9
Mg	279.8	Vegetable Oil	Level 3		103	4.57	4.44	9.8	3	0.0928	3.05	0.462	0.0222	181	0.0514	0.171	10.5	34.9

Mn	257.61	Breaded Chicken	Level 1		99.1	14.2	14.3	0.98	3	2.55	0.0467	0.0254	0.00517	2.1	0.00208	0.00694	0.425	1.42
Mn	257.61	Breaded Chicken	Level 2		98.7	8.74	8.86	0.98	3	2.55	0.0467	0.0328	0.00113	3.96	0.00208	0.00694	0.425	1.42
Mn	257.61	Breaded Chicken	Level 3		97	8.23	8.48	0.98	3	2.55	0.0467	0.0644	0.00311	9.91	0.00208	0.00694	0.425	1.42
Mn	257.61	Corn Flakes	Level 1	< LOD	49.2	25	50.8	0.98	3	-0.115	0.0812	0.000696	0.000359	0.284	0.00208	0.00694	0.425	1.42
Mn	257.61	Corn Flakes	Level 2	< LOD	60.4	7.48	12.4	0.98	3	-0.115	0.0812	0.00183	0.000319	0.573	0.00208	0.00694	0.425	1.42
Mn	257.61	Corn Flakes	Level 3	Trace	80.1	0.683	0.853	0.98	3	-0.115	0.0812	0.00572	0.000116	1.42	0.00208	0.00694	0.425	1.42
Mn	257.61	Italian Dressing	Level 1	< LOD	-72.8	28.2	-38.7	0.98	3	-0.278	0.0855	0.000687	0.000246	0.193	0.00208	0.00694	0.425	1.42
Mn	257.61	Italian Dressing	Level 2	< LOD	3.4	13.4	394	0.98	3	-0.278	0.0855	8.32E-05	0.000272	0.423	0.00208	0.00694	0.425	1.42
Mn	257.61	Italian Dressing	Level 3	Trace	66.3	6.95	10.5	0.98	3	-0.278	0.0855	0.00338	0.000405	0.987	0.00208	0.00694	0.425	1.42
Mn	257.61	Vegetable Oil	Level 1	< LOD	-21.3	31.6	-148	0.98	3	-0.683	0.041	0.000354	0.000555	0.677	0.00208	0.00694	0.425	1.42
Mn	257.61	Vegetable Oil	Level 2	< LOD	49.1	8.77	17.9	0.98	3	-0.683	0.041	0.00181	0.000336	1.42	0.00208	0.00694	0.425	1.42
Mn	257.61	Vegetable Oil	Level 3		80.5	5.44	6.76	0.98	3	-0.683	0.041	0.00812	0.000594	4.07	0.00208	0.00694	0.425	1.42
Na	588.995	Breaded Chicken	Level 1		87.8	18.8	21.4	49	3	7040	158	56.5	9.4	3750	0.538	1.79	110	366
Na	588.995	Breaded Chicken	Level 2		88.9	7.05	7.93	49	3	7040	158	68	2.79	7140	0.538	1.79	110	366
Na	588.995	Breaded Chicken	Level 3		98.3	13.6	13.8	49	3	7040	158	130	5.33	17800	0.538	1.79	110	366
Na	588.995	Corn Flakes	Level 1		87.7	4.47	5.1	49	3	7310	83.3	50.4	2.41	3300	0.538	1.79	110	366
Na	588.995	Corn Flakes	Level 2		88	1.54	1.75	49	3	7310	83.3	69.3	3.44	6680	0.538	1.79	110	366
Na	588.995	Corn Flakes	Level 3		92	3.18	3.46	49	3	7310	83.3	113	4.01	16500	0.538	1.79	110	366
Na	588.995	Italian Dressing	Level 1		103	5.17	5.02	49	3	11300	91	79.7	1.64	4680	0.538	1.79	110	366
Na	588.995	Italian Dressing	Level 2		94.4	7.64	8.09	49	3	11300	91	104	1.11	10600	0.538	1.79	110	366
Na	588.995	Italian Dressing	Level 3		99.3	3.7	3.73	49	3	11300	91	186	0.178	24800	0.538	1.79	110	366
Na	588.995	Vegetable Oil	Level 1		78.2	36.8	47.1	49	3	740	21.4	2.1	0.0614	97.9	0.538	1.79	110	366
Na	588.995	Vegetable Oil	Level 2		89.7	25.9	28.9	49	3	740	21.4	2.4	0.073	206	0.538	1.79	110	366
Na	588.995	Vegetable Oil	Level 3		101	5.3	5.25	49	3	740	21.4	3.31	0.113	589	0.538	1.79	110	366
P	213.618	Breaded Chicken	Level 1		98.9	19.2	19.4	9.8	3	1540	25.6	13.3	2.17	898	0.0287	0.0957	5.86	19.5
P	213.618	Breaded Chicken	Level 2		97	8.67	8.94	9.8	3	1540	25.6	16.2	0.578	1710	0.0287	0.0957	5.86	19.5
P	213.618	Breaded Chicken	Level 3		94.9	6.71	7.07	9.8	3	1540	25.6	29.7	1.57	4270	0.0287	0.0957	5.86	19.5
P	213.618	Corn Flakes	Level 1		99	3.66	3.7	9.8	3	579	4.4	4.08	0.192	249	0.0287	0.0957	5.86	19.5
P	213.618	Corn Flakes	Level 2		96.2	1.12	1.16	9.8	3	579	4.4	5.58	0.308	502	0.0287	0.0957	5.86	19.5
P	213.618	Corn Flakes	Level 3		97.9	2.34	2.39	9.8	3	579	4.4	9.04	0.263	1240	0.0287	0.0957	5.86	19.5
P	213.618	Italian Dressing	Level 1		93.1	14.3	15.4	9.8	3	98.6	1.61	0.741	0.0424	55.4	0.0287	0.0957	5.86	19.5
P	213.618	Italian Dressing	Level 2		84.9	5.28	6.22	9.8	3	98.6	1.61	0.988	0.0148	122	0.0287	0.0957	5.86	19.5

P	213.618	Italian Dressing	Level 3		93.8	9.34	9.96	9.8	3	98.6	1.61	1.88	0.158	284	0.0287	0.0957	5.86	19.5
P	213.618	Vegetable Oil	Level 1	< LOD	-44.1	8.72	-19.8	9.8	3	-27.1	2.35	-0.023	0.00339	20.4	0.0287	0.0957	5.86	19.5
P	213.618	Vegetable Oil	Level 2	< LOD	24.2	9.37	38.7	9.8	3	-27.1	2.35	0.0268	0.0104	42.8	0.0287	0.0957	5.86	19.5
P	213.618	Vegetable Oil	Level 3		70.5	4.59	6.51	9.8	3	-27.1	2.35	0.214	0.0169	123	0.0287	0.0957	5.86	19.5
S	181.972	Breaded Chicken	Level 1		97.2	21.8	22.4	9.8	3	1490	15.1	13	1.97	922	0.0665	0.222	13.6	45.2
S	181.972	Breaded Chicken	Level 2		97	6.69	6.9	9.8	3	1490	15.1	16.2	0.735	1750	0.0665	0.222	13.6	45.2
S	181.972	Breaded Chicken	Level 3		92.3	5.96	6.46	9.8	3	1490	15.1	29.3	1.33	4380	0.0665	0.222	13.6	45.2
S	181.972	Corn Flakes	Level 1		95.6	7.13	7.46	9.8	3	656	9.83	4.76	0.263	321	0.0665	0.222	13.6	45.2
S	181.972	Corn Flakes	Level 2		92.6	1.13	1.22	9.8	3	656	9.83	6.6	0.36	647	0.0665	0.222	13.6	45.2
S	181.972	Corn Flakes	Level 3		94.3	1.07	1.13	9.8	3	656	9.83	10.9	0.249	1600	0.0665	0.222	13.6	45.2
S	181.972	Italian Dressing	Level 1		107	14.1	13.2	9.8	3	182	2.57	1.39	0.0711	93.3	0.0665	0.222	13.6	45.2
S	181.972	Italian Dressing	Level 2		100	8.6	8.6	9.8	3	182	2.57	1.9	0.025	205	0.0665	0.222	13.6	45.2
S	181.972	Italian Dressing	Level 3		104	4.48	4.31	9.8	3	182	2.57	3.51	0.154	478	0.0665	0.222	13.6	45.2
S	181.972	Vegetable Oil	Level 1	Trace	85.2	14.3	16.8	9.8	3	15	3.4	0.125	0.0073	39.4	0.0665	0.222	13.6	45.2
S	181.972	Vegetable Oil	Level 2		86.8	6.98	8.04	9.8	3	15	3.4	0.225	0.0136	82.7	0.0665	0.222	13.6	45.2
S	181.972	Vegetable Oil	Level 3		94.3	3.31	3.51	9.8	3	15	3.4	0.59	0.0352	237	0.0665	0.222	13.6	45.2
Zn	202.548	Breaded Chicken	Level 1		98.8	14.9	15.1	0.98	3	5.56	0.0636	0.0506	0.00958	3.71	0.00193	0.00642	0.393	1.31
Zn	202.548	Breaded Chicken	Level 2		101	8.92	8.83	0.98	3	5.56	0.0636	0.0642	0.00268	7	0.00193	0.00642	0.393	1.31
Zn	202.548	Breaded Chicken	Level 3		97.3	7.34	7.54	0.98	3	5.56	0.0636	0.12	0.00467	17.5	0.00193	0.00642	0.393	1.31
Zn	202.548	Corn Flakes	Level 1		94.3	20.8	22.1	0.98	3	1.27	0.21	0.00996	0.00115	0.784	0.00193	0.00642	0.393	1.31
Zn	202.548	Corn Flakes	Level 2		95	8.26	8.69	0.98	3	1.27	0.21	0.0146	0.0011	1.58	0.00193	0.00642	0.393	1.31
Zn	202.548	Corn Flakes	Level 3		97	1.17	1.21	0.98	3	1.27	0.21	0.0255	0.000155	3.91	0.00193	0.00642	0.393	1.31
Zn	202.548	Italian Dressing	Level 1		108	4.52	4.19	0.98	3	-0.112	0.144	0.0515	0.00123	9.63	0.00193	0.00642	0.393	1.31
Zn	202.548	Italian Dressing	Level 2		104	8.62	8.29	0.98	3	-0.112	0.144	0.107	0.00193	21.2	0.00193	0.00642	0.393	1.31
Zn	202.548	Italian Dressing	Level 3		106	1.92	1.81	0.98	3	-0.112	0.144	0.27	0.00532	49.4	0.00193	0.00642	0.393	1.31
Zn	202.548	Vegetable Oil	Level 1	Trace	61.4	3.02	4.92	0.98	3	-0.843	0.473	0.00414	0.000414	2.61	0.00193	0.00642	0.393	1.31
Zn	202.548	Vegetable Oil	Level 2		79.5	3.36	4.23	0.98	3	-0.843	0.473	0.0113	0.000468	5.49	0.00193	0.00642	0.393	1.31
Zn	202.548	Vegetable Oil	Level 3		93.3	2.36	2.53	0.98	3	-0.843	0.473	0.0363	0.00155	15.7	0.00193	0.00642	0.393	1.31

## R Code

Prepare working environment

```
library(tidyverse)
library(lubridate)
library(janitor)
library(ggpmisc)
library(readxl)
library(broom)
library(viridis)
library(scales)
library(ggbeeswarm)
library(patchwork)
library(ggforce)
library(gt)
```

```
my.formula <- y ~ x
```

```
theme_master <- function(base_size=14) {
  library(grid)
  (theme_bw(base_size = base_size)+
    theme(text=element_text(color="black"),
          axis.title=element_text(size = rel(1.3)),
          axis.text=element_text(size = rel(0.8), color = "black"),
          axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 0.5),
          legend.background=element_rect(fill="transparent"),
          legend.key.size = unit(1, 'lines'),
          strip.text = element_text(size = rel(0.8), color = "black"),
          panel.border=element_rect(color="black",linewidth=1)
    ))
}
```

```
theme_set(theme_master())
```

Single wavelengths for elements of interest

```
single_line <-
c(
  "588.995",
  "766.491",
  "279.800",
  "315.887",
  "257.610",
  "238.204",
  "327.395",
  "202.548",
  "213.618",
  "181.972"
)
```

Reading in data

Reading in supplemental info containing data from reference material certificate of analysis

```

ref_material_info <-
  c("NIST_3233_info",
    "NIST_1845a_info",
    "NIST_1849a_info",
    "NIST_1549a_info",
    "NIST_1577c_info",
    "NIST_1566b_info",
    "NIST_1568b_info",
    "FDA_cocoa_info",
    "3290_info",
    "NMIJ_7405b_info",
    "NRC_dolt-4_info",
    "NIST_2976_info",
    "NRC_dorm-4_info")

RM_info <-
  lst()

RM_info <- lapply(seq_along(ref_material_info), function(x) {
  RM_info[[x]] <-
    read_excel("C:/Users/Jake.Carter/OneDrive - FDA/HomeDrive/Data/ICP
OES/SLV/summary/SLV_summary.xlsx",
              sheet = ref_material_info[x]) %>%
  dplyr::mutate(QC = "Reference material",
               ref_material = ref_material_info[x],
               ref_material = str_replace(ref_material, "_info", ""),
               ref_material = str_replace(ref_material, "NIST_1845a", "1845a"),
               ref_material = str_replace(ref_material, "NIST_1549a", "1549a"),
               ref_material = str_replace(ref_material, "NIST_1849a", "1849a"),
               ref_material = str_replace(ref_material, "NIST_3233", "3233"),
               ref_material = str_replace(ref_material, "NIST_1577c", "1577c"),
               ref_material = str_replace(ref_material, "NIST_1566b", "1566b"),
               ref_material = str_replace(ref_material, "NIST_1568b", "1568b"),
               ref_material = str_replace(ref_material, "NIST_2976", "2976"),
               ref_material = str_replace(ref_material, "NRC_dolt-4", "dolt-4"),
               ref_material = str_replace(ref_material, "NRC_dorm-4", "dorm-4"),
               ref_material = str_replace(ref_material, "FDA_cocoa", "FDA cocoa"))
})

names(RM_info) <- ref_material_info

RM_info <-
  dplyr::bind_rows(RM_info) %>%
  dplyr::rename(certificate_value = certificate_value_mg_kg,
               sample = ref_material) %>%
  dplyr::mutate(QC = "Reference Material")

Read in dilution data

dilutions <-
  read_excel("C:/Users/Jake.Carter/OneDrive - FDA/HomeDrive/Data/ICP
OES/2022/June/20220606/SDA.xlsx",
            sheet = "R") %>%
  dplyr::mutate(replicate = as.character(replicate)) %>%

```

```
tidyr::pivot_longer(cols = Al:Zn, values_to = "c_std", names_to = "symbol") %>%
dplyr::mutate(date = as.character(date))
```

### Dry mass conversion factors for reference material analysis

```
dry_mass <-
read_excel("C:/Users/Jake.Carter/OneDrive - FDA/HomeDrive/Data/ICP
OES/2022/June/20220606/SDA.xlsx",
sheet = "dry_mass") %>%
dplyr::rename(sample = ref_mat) %>%
dplyr::mutate(date = lubridate::ymd(date))
```

Read in fortification dilutions Spike conc. is  $\frac{C_s M_s}{M_x}$

```
FAPs <-
read_excel("C:/Users/Jake.Carter/OneDrive - FDA/HomeDrive/Data/ICP
OES/2022/June/20220606/SDA.xlsx",
sheet = "FAPs") %>%
tidyr::pivot_longer(cols = As:Zn, names_to = "symbol", values_to = "spike_conc") %>%
dplyr::mutate(replicate = as.character(replicate),
date = as.character(date))
```

### Read in data

In this experiment, Lu is in solution one (S1) and is internal standard 1 (IS1). In is in solution two (S2) and is internal standard 2 (IS2).

```
files = list.files(path = "C:/Users/Jake.Carter/OneDrive - FDA/HomeDrive/Data/ICP
OES/2022/June/20220606", pattern="*.csv")
```

```
data <-
readr::read_csv(paste("C:/Users/Jake.Carter/OneDrive - FDA/HomeDrive/Data/ICP
OES/2022/June/20220606/", files[1], sep = ""),
skip = 6) %>%
janitor::clean_names() %>%
dplyr::select(label, element, element_label, date_time, starts_with("intensity_replicate")) %>%
tidyr::pivot_longer(cols = intensity_replicate_1:intensity_replicate_100,
values_to = "intensity",
names_to = "intensity_replicate") %>%
dplyr::mutate(intensity_replicate = str_remove(intensity_replicate, "intensity_replicate_"),
element = str_replace(element, " ", "."),
label = str_replace(label, "d-blk", "blk"),
label = str_replace(label, "fda_cocoa_cp_35", "FDA cocoa"),
date = str_extract(label, "2022-03-09|2022-04-07|2022-02-01"),
label = str_remove(label, "2022-03-09-|2022-04-07-|2022-02-01-"),
sample = str_remove(label, "-10$|-1$|-2$|-3$|-4$|-5$|-6$|-7$|-8$|-9$"),
replicate = str_extract(label, "1$|2$|3$|4$|5$|6$|7$|8$|9$|10$"),
FAP = str_extract(label, "FAP1|FAP2|FAP3"),
FAP = if_else(is.na(FAP),
"Not an FAP",
FAP),
sample = str_remove(sample, "-FAP1|-FAP2|-FAP3"),
QC = if_else(FAP %in% c("FAP1", "FAP2", "FAP3"),
"FAP",
if_else(sample %in% RM_info$sample,
"Reference Material",
```

```

      if_else(sample == "blk",
              "blank",
              "UAP")
    )
  ) %>%
dplyr::rename(symbol = element_label) %>%
dplyr::group_by(element, sample, date_time, replicate) %>%
dplyr::mutate(normalized = intensity/max(intensity)) %>%
dplyr::ungroup() %>%
dplyr::filter(str_detect(label, "rinse", negate = TRUE))

```

```

samples <-
  unique(data$label)

```

```

lines <-
  unique(data$element)

```

We plot raw data and generate linear regressions. Model parameters will be used to determine solution concentrations according to automatic standard dilution analysis

```

sda1 <-
  data %>%
  dplyr::select(-intensity) %>%
  dplyr::filter(element %in% c("Zn.202.548", "In.325.609", "Lu.261.541"),
                str_detect(label, "1577c", negate = FALSE)) %>%
  dplyr::mutate(replicate = paste("Rep. #", replicate, sep = ""),
                replicate = fct_inorder(replicate),
                dig_label = str_replace(date, "2022-02-01", "Dig. 1"),
                dig_label = str_replace(dig_label, "2022-03-09", "Dig. 2"),
                dig_label = str_replace(dig_label, "2022-04-07", "Dig. 3"),
                element = str_replace(element, "In.", "In "),
                element = str_replace(element, "Lu.", "Lu "),
                element = str_replace(element, "Zn.", "Zn "),
                element = paste(element, "(nm)")) %>%
  ggplot(aes(x = as.numeric(intensity_replicate), y = normalized, fill = element, shape = element)) +
  geom_point(size = 1,
             stroke = 0.2) +
  scale_shape_manual(values = c(21,22,23)) +
  guides(fill = guide_legend(override.aes = list(size=2))) +
  scale_fill_viridis_d(option = "turbo", alpha = 0.5) +
  labs(y = "Normalized Intensity",
       x = "Intensity Replicate",
       subtitle = "(a)",
       fill = NULL,
       shape = NULL) +
  facet_grid(dig_label~replicate) +
  theme_master(base_size = 7.5) +
  theme(legend.position = "right",
        panel.grid = element_blank())

```

```

sda1

```

```

sda2 <-
  data %>%
  dplyr::select(-normalized, -symbol) %>%

```

```

dplyr::filter(element %in% c("In.325.609", "Lu.261.541")) %>%
dplyr::mutate(element = str_replace(element, "In.", "In "),
  element = str_replace(element, "Lu.", "Lu "),
  element = str_replace(element, "Zn.", "Zn "),
  element = paste(element, "(nm)")) %>%
tidyr::pivot_wider(names_from = element, values_from = intensity) %>%
dplyr::filter(str_detect(label, "1577c")) %>%
dplyr::mutate(replicate = paste("Rep. #", replicate, sep = ""),
  replicate = fct_inorder(replicate),
  dig_label = str_replace(date, "2022-02-01", "Dig. 1"),
  dig_label = str_replace(dig_label, "2022-03-09", "Dig. 2"),
  dig_label = str_replace(dig_label, "2022-04-07", "Dig. 3")) %>%
ggplot(aes(x = `In 325.609 (nm)`/10^4, y = `Lu 261.541 (nm)`/10^4)) +
geom_point(size = 0.75,
  stroke = 0.2,
  alpha = 0.8,
  shape = 21,
  fill = "gray") +
geom_smooth(method = "lm",
  se = FALSE,
  linewidth = 0.3,
  formula = my.formula,
  color = plasma(1, alpha = 0.75, direction = -1, end = 0.8)) +
stat_poly_eq(formula = my.formula,
  aes(label = paste(..eq.label.., sep = "~~~")),
  size = rel(1.6),
  label.y = c(rep(0.95, 27)),
  label.x = c(rep(0.95, 27)),
  parse = TRUE) +
labs(subtitle = "(b)",
  y = expression(paste("Lu 261.541 (nm) / ", 10^4)),
  x = expression(paste("In 325.609 (nm) / ", 10^4))) +
theme(legend.position = "none") +
facet_grid(dig_label~replicate) +
theme_master(base_size = 7.5) +
theme(panel.grid = element_blank())

```

sda2

sda3 <-

```

data %>%
dplyr::select(-normalized, -symbol) %>%
tidyr::pivot_wider(names_from = element, values_from = intensity) %>%
tidyr::pivot_longer(cols = -starts_with(c("intensity_replicate",
  "sample",
  "date_time",
  "FAP",
  "QC",
  "replicate",
  "date",
  "In",
  "Lu",
  "label")), names_to = "element", values_to = "intensity") %>%
dplyr::filter(element == "Zn.202.548") %>%
dplyr::filter(str_detect(label, "1577c")) %>%
dplyr::mutate(replicate = paste("Rep. #", replicate, sep = "")),

```

```

    replicate = fct_inorder(replicate),
    dig_label = str_replace(date, "2022-02-01", "Dig. 1"),
    dig_label = str_replace(dig_label, "2022-03-09", "Dig. 2"),
    dig_label = str_replace(dig_label, "2022-04-07", "Dig. 3")) %>%
ggplot(aes(x = Lu.261.541/10^4, y = intensity/10^4)) +
geom_point(size = 0.75,
    stroke = 0.2,
    alpha = 0.8,
    shape = 21,
    fill = "gray") +
geom_smooth(method = "lm",
    se = FALSE,
    formula = my.formula,
    linewidth = 0.3,
    color = plasma(1, alpha = 0.75, direction = -1, end = 0.8)) +
stat_poly_eq(formula = my.formula,
    aes(label = paste(..eq.label.., sep = "~~~")),
    size = rel(1.6),
    label.y = c(rep(0.95, 24)),
    parse = TRUE) +
labs(y = expression(paste("Zn 202.548 (nm) / ", 10^4)),
    x = expression(paste("Lu 261.541 (nm) / ", 10^4)),
    subtitle = "(c)") +
theme(legend.position = "none",
    axis.text.x = element_text(angle = 90, vjust = 0.5)) +
facet_grid(dig_label~replicate,
    scales = "free") +
theme_master(base_size = 7.5) +
theme(panel.grid = element_blank())

```

sda3

Code for data vis

sda1/(sda2 + sda3)

Determining automatic SDA parameters

From the text: Solution concentrations were determined according to eq. 1,<sup>1</sup> where  $C_A^{soln}$ ,  $C_A^{std}$ , and  $S_I^{max}$ , is the concentration of the analyte in the sample solution after digestion, the concentration of the analyte in the standard solution, and the maximum signal of IS1, respectively. The intercept and slope in eq. 1 were determined from the linear regression model generated from plotting the signal of the analyte against the signal of IS1.  $S_I^{max}$  was determined from plotting the signal of IS1 against the signal of IS2, fitting a linear regression model, and generating the y-intercept.<sup>1</sup>

$$\text{eq. 1: } C_A^{soln} = \frac{\text{intercept}}{\text{slope}} * \frac{C_A^{std}}{S_I^{max}}$$

Now we compute linear regression parameters (e.g. intercept\_1, intercept\_2, and slope) in batch. Here, intercept\_2 is the same as  $S_I^{max}$

```

line_1_metrics <-
data %>%
dplyr::select(-normalized, -symbol) %>%
tidyr::pivot_wider(names_from = element, values_from = intensity) %>%

```

```

tidyr::pivot_longer(cols = -starts_with(c("label",
      "replicate",
      "sample",
      "date",
      "date_time",
      "QC",
      "FAP",
      "In",
      "Lu")), names_to = "element", values_to = "intensity") %>%
dplyr::rename(intstd_1 = Lu.261.541) %>%
dplyr::select(-starts_with(c("In.", "Lu"))) %>%
tidyr::nest(data = c("intensity_replicate", "intstd_1", "intensity")) %>%
mutate(
  fit = map(data, ~ lm(intensity ~ intstd_1, data = .x)),
  tidied = map(fit, tidy)
) %>%
unnest(tidied) %>%
dplyr::select(-fit)

intercept_1 <-
line_1_metrics %>%
dplyr::filter(term == "(Intercept)") %>%
dplyr::rename(metric = estimate) %>%
dplyr::select(label, element, metric, sample, replicate, FAP, QC, date, date_time) %>%
dplyr::mutate(term = "intercept_1")

slope_1 <-
line_1_metrics %>%
dplyr::filter(term == "intstd_1") %>%
dplyr::rename(metric = estimate) %>%
dplyr::select(label, element, metric, sample, replicate, FAP, QC, date, date_time) %>%
dplyr::mutate(term = "slope_1")

intstd_metrics <-
data %>%
dplyr::select(-normalized, -symbol) %>%
tidyr::pivot_wider(names_from = element, values_from = intensity) %>%
tidyr::pivot_longer(cols = -starts_with(c("intensity_replicate",
      "replicate",
      "sample",
      "date",
      "date_time",
      "QC",
      "FAP",
      "In",
      "Lu",
      "label")), names_to = "element", values_to = "intensity") %>%
dplyr::rename(intstd_1 = Lu.261.541,
  intstd_2 = In.325.609) %>%
dplyr::select(-starts_with(c("In.", "Lu")), -intensity) %>%
distinct() %>%
tidyr::nest(data = c("intensity_replicate", "intstd_1", "intstd_2")) %>%
mutate(
  fit = map(data, ~ lm(intstd_1 ~ intstd_2, data = .x)),
  tidied = map(fit, tidy)
)

```

```
) %>%  
unnest(tidied)
```

```
intercept_2 <-  
  intstd_metrics %>%  
  dplyr::filter(term == "(Intercept)") %>%  
  dplyr::rename(metric = estimate) %>%  
  dplyr::select(label, element, metric, sample, replicate, FAP, QC, date, date_time) %>%  
  dplyr::mutate(term = "intercept_2")
```

Combine all relevant parameters into one data tibble

```
sda_metrics <-  
  intercept_1 %>%  
  dplyr::bind_rows(slope_1, intercept_2) %>%  
  tidyr::pivot_wider(names_from = term, values_from = metric) %>%  
  dplyr::mutate(symbol = str_remove(element, "[:digit:]+[:digit:]+")  
  ) %>%  
  dplyr::left_join(dilutions) %>%  
  dplyr::mutate(c_soln = (intercept_1/slope_1) * (c_std/intercept_2),  
    c_sample = c_soln * dil_samp_factor) %>%  
  dplyr::arrange(element, label) %>%  
  dplyr::left_join(RM_info)
```

Blank analysis

One of the blanks has high Fe. More than likely used wrong solution or severe contamination

```
data %>%  
  dplyr::select(-intensity) %>%  
  dplyr::filter(element %in% c("Fe.238.204", "In.325.609", "Lu.261.541"),  
    str_detect(label, "blk", negate = FALSE)) %>%  
  dplyr::mutate(replicate = paste("Rep. #", replicate, sep = ""),  
    replicate = fct_inorder(replicate)) %>%  
  ggplot(aes(x = as.numeric(intensity_replicate), y = normalized, fill = element)) +  
  geom_point(alpha = 0.5,  
    shape = 21,  
    size = 1) +  
  scale_fill_viridis_d(option = "turbo", alpha = 0.5) +  
  labs(y = "Normalized intensity",  
    x = "Intensity Replicate",  
    fill = NULL) +  
  facet_grid(date~replicate) +  
  theme_master(base_size = 6) +  
  theme(legend.position = "top")
```

Determining figures of merit

```
FOM <-  
  sda_metrics %>%  
  dplyr::mutate(remove = if_else(  
    sample == "blk" & replicate == 1 & date == "2022-04-07" & symbol == "Fe",  
    "remove",  
    "keep"  
  )) %>%  
  dplyr::filter(sample == "blk",  
    remove == "keep") %>%
```

```

dplyr::group_by(element) %>%
dplyr::summarize(LOQ = 10*sd(c_sample),
  LOD = 3*sd(c_sample),
  ASDL = 3*sd(c_soln),
  ASQL = 10*sd(c_soln),
  n = n())

```

## Code for data processing

```

sda_metrics %>%
dplyr::mutate(remove = if_else(
  sample == "blk" & replicate == 1 & date == "2022-04-07" & symbol == "Fe",
  "remove",
  "keep"
)) %>%
dplyr::filter(sample == "blk",
  remove == "keep") %>%
dplyr::group_by(element) %>%
dplyr::summarize(LOQ = signif(10*sd(c_sample), 2),
  LOD = signif(3*sd(c_sample), 2),
  ASDL = signif(3*sd(c_soln), 2),
  ASQL = signif(10*sd(c_soln), 2),
  n = n()) %>%
dplyr::mutate(symbol = str_remove(element, "[:digit:]+[:digit:]+"),
  element = str_replace(element, "[^[:alpha:]]", ""),
  element = str_replace(element, "[:alpha:]", ""),
  element = str_replace(element, "[:alpha:]", "")) %>%
dplyr::filter(element %in% single_line) %>%
dplyr::mutate(element = paste(element, " nm", sep = "")) %>%
dplyr::rename(Element = symbol,
  line = element,
  `LOQ (mg/kg)` = LOQ,
  `LOD (mg/kg)` = LOD,
  `ASDL (mg/kg)` = ASDL,
  `ASQL (mg/kg)` = ASQL) %>%
dplyr::mutate(Element = paste(Element, line, sep = " ")) %>%
dplyr::select(Element, `ASDL (mg/kg)`, `ASQL (mg/kg)`, `LOD (mg/kg)`, `LOQ (mg/kg)`) %>%
gt()

```

## Joining FOM to data

```

data_out <-
data %>%
dplyr::group_by(label, element) %>%
dplyr::summarize(ratio = max(intensity)/min(intensity)) %>%
dplyr::ungroup() %>%
dplyr::right_join(sda_metrics,
  multiple = "all") %>%
dplyr::left_join(FOM %>%
  dplyr::select(-n)) %>%
dplyr::mutate(element = str_replace(element, "[^[:alpha:]]", ""),
  element = str_replace(element, "[:alpha:]", ""),
  element = str_replace(element, "[:alpha:]", ""),
  multiple = "all") %>%
dplyr::left_join(FAPs) %>%
dplyr::mutate(date = lubridate::ymd(date))

```

## Reference material analysis

### Reference material results summary

```
ref_mat_summary <-  
  data_out %>%  
  dplyr::filter(QC == "Reference Material") %>%  
  dplyr::mutate(rec = c_sample/certificate_value*100) %>%  
  dplyr::left_join(dry_mass) %>%  
  dplyr::group_by(element, sample, date) %>%  
  dplyr::summarize(symbol = symbol,  
    mean_conc = mean(c_sample),  
    sd_conc = sd(c_sample),  
    mean_conc_soln = mean(c_soln),  
    sd_conc_soln = sd(c_soln),  
    c_std = c_std,  
    n = n(),  
    certificate_value = certificate_value,  
    certified = certified,  
    coverage_factor = coverage_factor,  
    plus_minus = plus_minus,  
    mean_rec = mean(rec*dry_mass_correction),  
    sd_rec = sd(rec*dry_mass_correction),  
    sigma_2_rec = 2*sd(rec*dry_mass_correction),  
    dry_mass_correction = dry_mass_correction,  
    mean_ratio = mean(ratio),  
    mean_slope = mean(slope_1),  
    LOQ = LOQ,  
    LOD = LOD,  
    ASDL = ASDL,  
    ASQL = ASQL) %>%  
  dplyr::distinct() %>%  
  dplyr::mutate(  
    samp_uncert = (mean_conc * dry_mass_correction * 10)/100, # Set to 10 % nominal uncertainty  
    z_score = (mean_conc * dry_mass_correction - certificate_value)/(sqrt((samp_uncert^2) +  
    ((plus_minus/coverage_factor)^2)))  
  ) %>%  
  dplyr::arrange(element, sample) %>%  
  dplyr::ungroup()
```

### Solution concentrations plotted against certified values

```
refmat1 <-  
  ref_mat_summary %>%  
  dplyr::filter(mean_conc_soln > ASQL,  
    element %in% single_line,  
    !is.na(z_score),  
    certified == "certified",  
    mean_conc_soln/c_std < 10) %>%  
  ggplot(aes(x = certificate_value, y = (mean_conc * dry_mass_correction), fill = symbol, group = 1)) +  
  geom_point(size = 0.5,  
    shape = 21,  
    stroke = 0.2) +  
  geom_smooth(method = "lm",  
    se = FALSE,  
    formula = my.formula,
```

```

    linewidth = 0.25,
    lty = 2,
    show.legend = FALSE,
    color = plasma(1, alpha = 0.75, direction = -1, end = 0.8)) +
stat_poly_eq(formula = my.formula,
  aes(label = paste(..eq.label..., ..rr.label..., sep = "~~~")),
  size = rel(2),
  rr.digits = 3,
  label.x = 0.175,
  label.y = 0.945,
  parse = TRUE) +
geom_abline(slope = 1,
  size = 0.25) +
scale_fill_viridis_d(option = "turbo",
  alpha = 0.8) +
guides(shape = guide_legend(override.aes = list(size=2)),
  fill = "none") +
annotate("text", -Inf, Inf, label = "(a)", hjust = -0.2, vjust = 1.5) +
labs(y = "Found Conc. (mg/kg)",
  x = "Certified Conc. (mg/kg)",
  fill = NULL) +
theme_master(base_size = 8) +
theme(axis.text.x = element_text(angle = 0),
  legend.position = "none",
  panel.grid = element_blank(),
  plot.margin=unit(c(0,5,5,0),"pt"))

```

refmat1

## Z score summary

```

refmat2 <-
ref_mat_summary %>%
dplyr::filter(mean_conc_soln > ASQL,
  element %in% single_line,
  !is.na(z_score),
  certified == "certified",
  mean_conc_soln/c_std < 10) %>%
ggplot(aes(x = "All Elements", y = z_score, fill = symbol, group = 1)) +
geom_hline(yintercept = c(-2,2),
  size = 0.25) +
geom_hline(yintercept = 0,
  lty = 2,
  size = 0.25) +
geom_quasirandom(size = 0.5,
  shape = 21,
  stroke = 0.2,
  width = 0.25) +
geom_boxplot(width = 0.5,
  lwd = 0.2,
  show.legend = FALSE,
  outlier.shape = NA,
  fill = "transparent") +
scale_fill_viridis_d(alpha = 0.8,
  option = "turbo") +
guides(shape = guide_legend(override.aes = list(size=2)),

```

```

    fill = "none") +
  annotate("text", -Inf, Inf, label = "(b)", hjust = -0.2, vjust = 2) +
  labs(y = "Z Score",
       x = NULL,
       fill = NULL) +
  theme_master(base_size = 8) +
  theme(axis.text.x = element_text(angle = 0),
        axis.title.x = element_blank(),
        legend.position = "none",
        panel.grid = element_blank(),
        plot.margin=unit(c(0,0,0,0),"pt"))

```

refmat2

refmat3 <-

```

ref_mat_summary %>%
  dplyr::filter(mean_conc_soln > ASQL,
                element %in% single_line,
                !is.na(z_score),
                certified == "certified",
                mean_conc_soln/c_std < 10) %>%
  dplyr::group_by(element) %>%
  dplyr::mutate(n = n(),
                label = paste(symbol, " (n=", n, ")", sep = "")) %>%
  ggplot(aes(x = symbol, y = z_score, fill = symbol, group = symbol)) +
  geom_hline(yintercept = c(-2,2),
             size = 0.25) +
  geom_hline(yintercept = 0,
             lty = 2,
             size = 0.25) +
  geom_quasirandom(size = 0.5,
                  stroke = 0.2,
                  shape = 21,
                  width = 0.25) +
  geom_boxplot(width = 0.5,
              lwd = 0.2,
              show.legend = FALSE,
              outlier.shape = NA,
              fill = "transparent") +
  scale_fill_viridis_d(alpha = 0.8,
                       option = "turbo") +
  guides(shape = guide_legend(override.aes = list(size=2)),
         fill = "none") +
  annotate("text", -Inf, Inf, label = "(c)", hjust = -0.2, vjust = 2) +
  labs(y = "Z Score",
       x = NULL,
       fill = NULL) +
  theme_master(base_size = 8) +
  theme(axis.text.x = element_text(angle = 0),
        axis.title.x = element_blank(),
        legend.position = "none",
        panel.grid = element_blank(),
        plot.margin=unit(c(0,5,0,0),"pt"))

```

refmat3

What about variability

```
refmat4 <-
  ref_mat_summary %>%
  dplyr::filter(mean_conc_soln > ASQL,
    element %in% single_line,
    !is.na(z_score),
    certified == "certified",
    mean_conc_soln/c_std < 10) %>%
  dplyr::mutate(RSD = sd_conc/mean_conc * 100) %>%
  dplyr::group_by(element) %>%
  dplyr::mutate(n = n(),
    label = paste(symbol, " (n=", n, ")", sep = "")) %>%
  ggplot(aes(x = symbol, y = RSD, fill = symbol, group = symbol)) +
  geom_quasirandom(size = 0.5,
    stroke = 0.2,
    shape = 21,
    width = 0.25) +
  geom_boxplot(width = 0.5,
    lwd = 0.2,
    show.legend = FALSE,
    outlier.shape = NA,
    fill = "transparent") +
  geom_hline(yintercept = 10,
    size = 0.25) +
  scale_fill_viridis_d(alpha = 0.8,
    option = "turbo") +
  guides(shape = guide_legend(override.aes = list(size=2)),
    fill = "none") +
  annotate("text", -Inf, Inf, label = "(d)", hjust = -0.2, vjust = 1.5) +
  labs(y = "RSD (%)",
    x = NULL,
    fill = NULL) +
  theme_master(base_size = 8) +
  theme(axis.text.x = element_text(angle = 0),
    axis.title.x = element_blank(),
    legend.position = "none",
    panel.grid = element_blank(),
    plot.margin=unit(c(0,0,0,0),"pt"))
```

refmat4

Table format for Z score range. Removing outliers based on 1.5 interquartile range

```
ref_mat_summary %>%
  dplyr::filter(mean_conc_soln > ASQL,
    element %in% single_line,
    !is.na(z_score),
    certified == "certified",
    mean_conc_soln/c_std < 10) %>%
  dplyr::ungroup() %>%
  dplyr::mutate(IQR = IQR(z_score),
    percentile_25 = quantile(z_score, 0.25),
    percentile_75 = quantile(z_score, 0.75),
    min_threshold = percentile_25 - 1.5*IQR,
    max_threshold = percentile_75 + 1.5*IQR) %>%
```

```
#dplyr::filter(z_score >= min_threshold & z_score <= max_threshold) %>%
dplyr::summarize(`Z score min` = signif(min(z_score), 2),
  `Z score max` = signif(max(z_score), 2),
  `Median Z score` = signif(median(z_score), 2),
  n = n()) %>%
```

```
gt()
```

```
ref_mat_summary %>%
```

```
dplyr::filter(mean_conc_soln > ASQL,
  element %in% single_line,
  !is.na(z_score),
  certified == "certified",
  mean_conc_soln/c_std < 10) %>%
```

```
dplyr::ungroup() %>%
```

```
dplyr::mutate(IQR = IQR(z_score),
  percentile_25 = quantile(z_score, 0.25),
  percentile_75 = quantile(z_score, 0.75),
  min_threshold = percentile_25 - 1.5*IQR,
  max_threshold = percentile_75 + 1.5*IQR) %>%
```

```
dplyr::filter(z_score >= min_threshold & z_score <= max_threshold) %>%
```

```
dplyr::summarize(`Z score min` = signif(min(z_score), 2),
  `Z score max` = signif(max(z_score), 2),
  `Median Z score` = signif(median(z_score), 2),
  n = n()) %>%
```

```
gt()
```

Code for data vis

```
(refmat1 + refmat2)/(refmat3 + refmat4)
```

Fortified analytical portions (FAP)s analysis

Let's do FAP calculations

```
sample_metrics <-
```

```
data_out %>%
```

```
dplyr::filter(QC == "UAP") %>%
```

```
dplyr::group_by(element, sample, date) %>%
```

```
dplyr::summarize(symbol = symbol,
  mean_conc = mean(c_sample),
  sd_conc = sd(c_sample),
  mean_conc_soln = mean(c_soln),
  sd_conc_soln = sd(c_soln),
  c_std = mean(c_std),
  LOQ = LOQ,
  LOD = LOD,
  ASQL = ASQL,
  ASDL = ASDL)
```

eq 2:  $Recovery (\%) = \left( \frac{C_{x+s} - C_x}{\frac{C_s M_s}{M_x}} \right) * 100$ , where  $C_{x+s}$ ,  $C_x$ ,  $C_s$ ,  $M_s$ , and  $M_x$ , is the concentration

determined in the spiked sample, concentration determined in the unspiked sample, concentration of the spiking solution, mass of the spike solution, mass of the analytical sample portion, respectively

```

FAP_replicates <-
  data_out %>%
  dplyr::filter(!QC %in% c("Reference Material", "blank", "UAP")) %>%
  dplyr::left_join(sample_metrics,
    multiple = "all") %>%
  distinct() %>%
  dplyr::mutate(
    net_sample = if_else(
      mean_conc < LOD,
      c_sample,
      c_sample - mean_conc
    ),
    rec = net_sample/spike_conc*100
  ) %>%
  dplyr::arrange(symbol, element, label, FAP)

```

### Summary FAP data

```

FAP_replicates_summary <-
FAP_replicates %>%
  dplyr::filter(element %in% single_line) %>%
  dplyr::group_by(element, sample, FAP) %>%
  dplyr::summarize(mean_rec = mean(rec),
    sd_rec = sd(rec),
    sigma_2_rec = 2*sd(rec),
    n = n(),
    mean_c_sample = mean(c_sample),
    sd_c_sample = sd(c_sample),
    mean_conc = mean_conc,
    sd_conc = sd_conc,
    mean_spike_soln = mean(spike_conc),
    mean_conc_soln = mean(c_soln),
    sd_conc_soln = sd(c_soln),
    n = n(),
    c_std = c_std,
    LOD = LOD,
    LOQ = LOQ,
    ASDL = ASDL,
    ASQL = ASQL,
    symbol = symbol) %>%
  distinct() %>%
  dplyr::ungroup()

```

We plot results from individual replicates against the fortified concentrations

```

FAP1 <-
  FAP_replicates %>%
  dplyr::filter(element %in% single_line,
    c_soln > ASQL,
    c_soln/c_std < 10) %>%
  dplyr::mutate(n = n(),
    n = as.character(n)
  ) %>%
  ggplot(aes(x = spike_conc, y = net_sample, fill = symbol, group = 1)) +
  geom_point(size = 0.75,
    stroke = 0.2,

```

```

    shape = 21) +
geom_smooth(method = "lm",
  se = FALSE,
  formula = my.formula,
  linewidth = 0.25,
  lty = 2,
  show.legend = FALSE,
  color = plasma(1, alpha = 0.75, direction = -1, end = 0.8)) +
stat_poly_eq(formula = my.formula,
  aes(label = paste(..eq.label.., ..rr.label.., sep = "~~~")),
  size = rel(2),
  rr.digits = 3,
  label.x = 0.175,
  label.y = 0.945,
  parse = TRUE) +
geom_abline(slope = 1,
  size = 0.25) +
scale_fill_viridis_d(option = "turbo",
  alpha = 0.8) +
guides(shape = guide_legend(override.aes = list(size=2)),
  fill = "none") +
annotate("text", -Inf, Inf, label = "(a)", hjust = -0.2, vjust = 1.5) +
labs(y = "Found Conc. (mg/kg)",
  x = "Fortified Conc. (mg/kg)",
  fill = NULL) +
theme_master(base_size = 8) +
theme(axis.text.x = element_text(angle = 0),
  legend.position = "none",
  panel.grid = element_blank(),
  plot.margin=unit(c(0,5,5,0),"pt"))

```

FAP1

We look at the summary

```

FAP2 <-
FAP_replicates_summary %>%
dplyr::filter(mean_conc_soln > ASQL,
  mean_conc_soln/c_std < 10) %>%
ggplot(aes(x = "All Elements", y = mean_rec, fill = symbol, group = 1)) +
geom_quasirandom(width = 0.25,
  stroke = 0.2,
  shape = 21,
  size = 0.75) +
geom_boxplot(outlier.shape = NA,
  fill = "transparent",
  show.legend = FALSE,
  lwd = 0.25,
  width = 0.5) +
scale_fill_viridis_d(option = "turbo",
  alpha = 0.8) +
geom_hline(yintercept = 120,
  linetype = 1,
  size = 0.25) +
geom_hline(yintercept = 80,
  linetype = 1,

```

```

      size = 0.25) +
geom_hline(yintercept = 100,
           linetype = 2,
           size = 0.25) +
guides(shape = guide_legend(override.aes = list(size=2)),
       fill = "none") +
annotate("text", -Inf, Inf, label = "(b)", hjust = -0.2, vjust = 2) +
labs(x = NULL,
     fill = NULL,
     shape = NULL,
     y = "Mean Recovery (%)") +
theme_master(base_size = 8) +
theme(axis.text.x = element_text(angle = 0),
      legend.position = "none",
      panel.grid = element_blank(),
      plot.margin=unit(c(0,0,0,0),"pt"))

```

FAP2

Recoveries by element

FAP3 <-

```

FAP_replicates_summary %>%
dplyr::filter(mean_conc_soln/c_std < 10,
              mean_conc_soln > ASQL) %>%
dplyr::group_by(element) %>%
dplyr::mutate(n = n(),
              label = paste(symbol, " (n=", n, ")", sep = ""))
) %>%
ggplot(aes(x = symbol, y = mean_rec, fill = symbol, group = symbol)) +
geom_quasirandom(width = 0.25,
                 stroke = 0.2,
                 shape = 21,
                 size = 0.75) +
geom_boxplot(outlier.shape = NA,
             fill = "transparent",
             show.legend = FALSE,
             lwd = 0.25,
             width = 0.5) +
scale_fill_viridis_d(option = "turbo",
                     alpha = 0.8) +
geom_hline(yintercept = 120,
           linetype = 1,
           size = 0.25) +
geom_hline(yintercept = 80,
           linetype = 1,
           size = 0.25) +
geom_hline(yintercept = 100,
           linetype = 2,
           size = 0.25) +
guides(shape = guide_legend(override.aes = list(size=2)),
       fill = "none") +
annotate("text", -Inf, Inf, label = "(c)", hjust = -0.2, vjust = 2) +
labs(x = NULL,
     fill = NULL,
     shape = NULL,

```

```

y = "Mean Recovery (%)" +
theme_master(base_size = 8) +
theme(axis.text.x = element_text(angle = 0),
      legend.position = "none",
      panel.grid = element_blank(),
      plot.margin=unit(c(0,5,0,0),"pt"))

```

FAP3

Variability by element

```

FAP4 <-
FAP_replicates_summary %>%
dplyr::filter(mean_conc_soln/c_std < 10,
              mean_conc_soln > ASQL) %>%
dplyr::group_by(element) %>%
dplyr::mutate(n = n(),
              label = paste(symbol, " (n=", n, ")", sep = ""))
) %>%
ggplot(aes(x = symbol, y = sd_rec/mean_rec*100, fill = symbol, group = symbol)) +
geom_quasirandom(width = 0.25,
                 stroke = 0.2,
                 shape = 21,
                 size = 0.75) +
geom_boxplot(outlier.shape = NA,
             show.legend = FALSE,
             lwd = 0.25,
             fill = "transparent",
             width = 0.5) +
scale_fill_viridis_d(option = "turbo",
                    alpha = 0.8) +
geom_hline(yintercept = 10,
           linetype = 1,
           size = 0.25) +
guides(shape = guide_legend(override.aes = list(size=2)),
       fill = "none") +
annotate("text", -Inf, Inf, label = "(d)", hjust = -0.2, vjust = 1.5) +
labs(x = NULL,
     fill = NULL,
     shape = NULL,
     y = "RSD (%)") +
theme_master(base_size = 8) +
theme(axis.text.x = element_text(angle = 0),
      legend.position = "none",
      panel.grid = element_blank(),
      plot.margin=unit(c(0,0,0,0),"pt"))

```

FAP4

Code for data vis

(FAP1 + FAP2)/(FAP3 + FAP4)

Table format for FAP recovery range. Removing outliers based on 1.5 interquartile range

```

FAP_replicates_summary %>%
  dplyr::filter(mean_conc_soln/c_std < 10,
    mean_conc_soln > ASQL) %>%
  dplyr::ungroup() %>%
  dplyr::mutate(IQR = IQR(mean_rec),
    percentile_25 = quantile(mean_rec, 0.25),
    percentile_75 = quantile(mean_rec, 0.75),
    min_threshold = percentile_25 - 1.5*IQR,
    max_threshold = percentile_75 + 1.5*IQR) %>%
  #dplyr::filter(mean_rec >= min_threshold & mean_rec <= max_threshold) %>%
  dplyr::summarize(`Recovery min (%)` = signif(min(mean_rec), 2),
    `Recovery max (%)` = signif(max(mean_rec), 2),
    `Recovery mean (%)` = signif(mean(mean_rec), 2),
    `Mean +/- 2sigma (%)` = paste(signif(mean(mean_rec), 2), " +/- ", signif(2*sd(mean_rec), 2)),
    n = n()) %>%
  gt()

```

```

FAP_replicates_summary %>%
  dplyr::filter(mean_conc_soln/c_std < 10,
    mean_conc_soln > ASQL) %>%
  dplyr::ungroup() %>%
  dplyr::mutate(IQR = IQR(mean_rec),
    percentile_25 = quantile(mean_rec, 0.25),
    percentile_75 = quantile(mean_rec, 0.75),
    min_threshold = percentile_25 - 1.5*IQR,
    max_threshold = percentile_75 + 1.5*IQR) %>%
  dplyr::filter(mean_rec >= min_threshold & mean_rec <= max_threshold) %>%
  dplyr::summarize(`Recovery min (%)` = signif(min(mean_rec), 2),
    `Recovery max (%)` = signif(max(mean_rec), 2),
    `Recovery mean (%)` = signif(mean(mean_rec), 2),
    `Mean +/- 2sigma (%)` = paste(signif(mean(mean_rec), 2), " +/- ", signif(2*sd(mean_rec), 2)),
    n = n()) %>%
  gt()

```

## Determining automatic SDA working range

```

metrics_paramters <-
  sda_metrics %>%
  dplyr::filter(QC == "Reference Material") %>%
  dplyr::mutate(date = lubridate::ymd(date),
    element = str_replace(element, "[^[:alpha:]]", ""),
    element = str_replace(element, "[:alpha:]", ""),
    element = str_replace(element, "[:alpha:]", "")) %>%
  dplyr::bind_rows(
    FAP_replicates
  ) %>%
  dplyr::left_join(dry_mass) %>%
  dplyr::mutate(
    rec = if_else(QC == "Reference Material",
      (c_sample*dry_mass_correction)/certificate_value * 100,
      rec)
  ) %>%
  dplyr::group_by(date, sample, symbol, element) %>%
  dplyr::summarize(
    mean_rec = mean(rec),
    sd_rec = sd(rec),

```

```

mean_c_sample = mean(c_sample),
sd_c_sample = sd(c_sample),
mean_c_soln = mean(c_soln),
sd_c_soln = sd(c_soln),
c_std = mean(c_std),
mean_slope = mean(slope_1),
mean_intercept = mean(intercept_1),
n = n()
) %>%
dplyr::left_join(
  FOM %>%
dplyr::mutate(element = str_replace(element, "[^:alpha:]", ""),
              element = str_replace(element, "[:alpha:]", ""),
              element = str_replace(element, "[:alpha:]", "")) %>%
  dplyr::select(-n),
  by = "element"
) %>%
dplyr::mutate(ratio = mean_c_soln/c_std) %>%
tidyr::pivot_longer(cols = c(ratio, mean_intercept, mean_slope),
                    names_to = "metric",
                    values_to = "value") %>%
dplyr::mutate(rec_factor = if_else(
  mean_rec >= 80 & mean_rec <= 120,
  "accept",
  "decline"
))

metrics1 <-
metrics_paramters %>%
dplyr::filter(element %in% single_line,
              mean_c_soln > ASQL) %>%
tidyr::pivot_wider(names_from = "metric",
                  values_from = "value") %>%
ggplot(aes(mean_slope, mean_rec, fill = rec_factor, shape = rec_factor, size = rec_factor)) +
geom_point(stroke = 0.15) +
geom_linerange(aes(ymax = mean_rec + sd_rec,
                  ymin = mean_rec - sd_rec),
              size = 0.15) +
scale_shape_manual(values = c(21,22)) +
scale_size_manual(values=c(1, 1.4)) +
coord_cartesian(ylim = c(60, 160)) +
scale_fill_viridis_d(option = "turbo",
                    alpha = 0.5,
                    begin = 0.1,
                    end = 0.75) +
geom_hline(yintercept = c(80, 120),
           size = 0.25) +
geom_hline(yintercept = 100,
           lty = 2,
           size = 0.25) +
labs(y = "Mean Recovery +/- Stdev. (%)",
     x = expression(italic("Slope")),
     subtitle = "(d)",
     size = NULL,
     shape = NULL,
     color = NULL,

```

```

fill = NULL) +
theme_master(base_size = 9) +
theme(legend.position = "none",
      panel.grid = element_blank())

```

metrics1

```

metrics2 <-
metrics_paramters %>%
dplyr::filter(element %in% single_line,
              mean_c_soln > ASQL) %>%
tidyr::pivot_wider(names_from = "metric",
                  values_from = "value") %>%
ggplot(aes(mean_intercept, mean_rec, fill = rec_factor, shape = rec_factor, size = rec_factor)) +
geom_point(stroke = 0.15) +
coord_cartesian(ylim = c(60, 160)) +
geom_linerange(aes(ymin = mean_rec - sd_rec,
                  ymax = mean_rec + sd_rec,
                  size = 0.15) +
scale_size_manual(values=c(1, 1.4)) +
scale_shape_manual(values = c(21,22)) +
scale_fill_viridis_d(option = "turbo",
                    alpha = 0.5,
                    begin = 0.1,
                    end = 0.75) +
geom_hline(yintercept = c(80, 120),
           size = 0.25) +
geom_hline(yintercept = 100,
           lty = 2,
           size = 0.25) +
scale_x_log10() +
labs(y = "Mean Recovery +/- Stdev. (%)",
     x = expression(italic("Intercept")*" ("*Log[10]*" Scale)"),
     subtitle = "(c)",
     size = NULL,
     shape = NULL,
     fill = NULL) +
theme_master(base_size = 9) +
theme(legend.position = "none",
      panel.grid = element_blank())

```

metrics2

```

metrics3 <-
metrics_paramters %>%
dplyr::filter(element %in% single_line,
              mean_c_soln > ASQL) %>%
tidyr::pivot_wider(names_from = "metric",
                  values_from = "value") %>%
ggplot(aes(ratio, mean_rec, fill = rec_factor, shape = rec_factor, size = rec_factor)) +
geom_point(stroke = 0.25) +
coord_cartesian(ylim = c(60, 160)) +
scale_shape_manual(values = c(21,22)) +
geom_linerange(aes(ymin = mean_rec - sd_rec,
                  ymax = mean_rec + sd_rec,
                  size = 0.15) +

```

```

scale_size_manual(values=c(1, 1.4),
  labels = percent) +
scale_fill_viridis_d(option = "turbo",
  alpha = 0.5,
  begin = 0.1,
  end = 0.75) +
geom_hline(yintercept = c(80, 120),
  size = 0.25) +
geom_hline(yintercept = 100,
  lty = 2,
  size = 0.25) +
annotate("text", -Inf, Inf, label = "(a)", hjust = -0.2, vjust = 1.5) +
labs(y = "Mean Recovery +/- Stdev. (%)",
  x = NULL,
  size = NULL,
  shape = NULL,
  fill = NULL) +
theme_master(base_size = 12) +
theme(legend.position = "none",
  panel.grid = element_blank(),
  axis.title = element_blank(),
  plot.margin=unit(c(0,0,5,0),"pt"))

```

metrics3

```

metrics4 <-
metrics_paramters %>%
dplyr::filter(element %in% single_line,
  mean_c_soln > ASQL) %>%
tidyr::pivot_wider(names_from = "metric",
  values_from = "value") %>%
ggplot(aes(ratio, mean_rec, fill = rec_factor, shape = rec_factor, size = rec_factor)) +
geom_point(stroke = 0.25) +
geom_linerange(aes(ymin = mean_rec - sd_rec,
  ymax = mean_rec + sd_rec,
  size = 0.15) +
scale_size_manual(values=c(1, 1.4)) +
scale_shape_manual(values = c(21,22)) +
scale_fill_viridis_d(option = "turbo",
  alpha = 0.5,
  begin = 0.1,
  end = 0.75) +
geom_hline(yintercept = c(80, 120),
  size = 0.15) +
geom_hline(yintercept = 100,
  lty = 2,
  size = 0.15) +
coord_cartesian(xlim = c(0, 1),
  ylim = c(60, 140),
  expand = FALSE) +
scale_y_continuous(breaks = seq(60, 140, 40)) +
annotate("text", -Inf, Inf, label = "(b)", hjust = -0.2, vjust = 1.5) +
labs(y = "Mean Recovery +/- Stdev. (%)",
  x = expression("Soln. Conc./Std. Conc."),
  shape = NULL,
  size = NULL,

```

```

fill = NULL) +
theme_master(base_size = 12) +
theme(legend.position = "none",
      panel.grid = element_blank(),
      axis.title = element_blank(),
      plot.margin=unit(c(0,0,0,0),"pt"))

```

metrics4

Code for data vis

Edit according to peer reviewer

```

result <- metrics3/metrics4
gt <- patchwork::patchworkGrob(result)

```

plot\_for\_save <-

```

gridExtra::grid.arrange(gt, left = "Mean Recovery +/- Stdev. (%)", bottom = "Soln. Conc./Std. Conc.")

```

Code for data vis

```

data %>%
dplyr::select(-intensity) %>%
dplyr::filter(element %in% c("Zn.202.548", "In.325.609", "Lu.261.541"),
              str_detect(label, "1566b|1577c|veg-oil-FAP1", negate = FALSE)) %>%
dplyr::mutate(replicate = paste("Rep. #", replicate, sep = ""),
              replicate = fct_inorder(replicate),
              dig_label = str_replace(date, "2022-02-01", "Dig. 1"),
              dig_label = str_replace(dig_label, "2022-03-09", "Dig. 2"),
              dig_label = str_replace(dig_label, "2022-04-07", "Dig. 3"),
              element = str_replace(element, "In.", "In "),
              element = str_replace(element, "Lu.", "Lu "),
              element = str_replace(element, "Zn.", "Zn "),
              element = paste(element, "(nm)")
              ) %>%
dplyr::filter(dig_label == "Dig. 2",
              replicate == "Rep. #1") %>%
ggplot(aes(x = as.numeric(intensity_replicate), y = normalized, fill = element, shape = element)) +
geom_point(size = 1,
           stroke = 0.1) +
scale_shape_manual(values = c(21,22,23)) +
annotate("text", x = 50.5, y = 0.85, label = "NIST 1566b", size = 1.5) +
annotate("text", x = 50.5, y = 0.52, label = "NIST 1577c", size = 1.5) +
annotate("text", x = 50.5, y = 0.055, label = "Veg. Oil FAP", size = 1.5) +
annotate("rect", xmin = 24.5, xmax = 43.5, ymin = 0, ymax = 1,
         alpha = .25,fill = "gray") +
annotate("rect", xmin = 58.5, xmax = 77.5, ymin = 0, ymax = 1,
         alpha = .25,fill = "gray") +
annotate("rect", xmin = 94.5, xmax = 100.5, ymin = 0, ymax = 1,
         alpha = .25,fill = "gray") +
annotate("segment", x = 18, xend = 25, y = 0.5, yend = 0.5, size = 0.25,) +
annotate("text", x = 10, y = 0.505, label = "SDA region", size = 2) +
scale_fill_viridis_d(option = "turbo", alpha = 0.5) +
guides(fill = guide_legend(override.aes = list(size=2))) +
#geom_hline(yintercept = 0.75) +
labs(y = "Normalized Intensity",

```

```

x = "Intensity Replicate",
fill = NULL,
shape = NULL) +
theme_master(base_size = 8) +
theme(legend.position = "top",
      panel.grid = element_blank(),
      legend.title=element_blank(),
      legend.margin = margin(0, 0, 0, 0),
      legend.spacing.x = unit(0, "mm"),
      legend.spacing.y = unit(0, "mm"),
      plot.margin=unit(c(0,0,0,0),"pt"))

working_range <-
ref_mat_summary %>%
dplyr::filter(element %in% single_line,
              mean_conc_soln > ASQL) %>%
dplyr::select(element,
              symbol,
              sample,
              mean_conc,
              sd_conc,
              mean_rec,
              sd_rec,
              sigma_2_rec,
              mean_conc_soln,
              c_std,
              LOD,
              LOQ,
              ASDL,
              ASQL) %>%
dplyr::rename(mean_c_sample = mean_conc,
              sd_c_sample = sd_conc) %>%
dplyr::mutate(QC = "Reference Material") %>%
dplyr::bind_rows(
  FAP_replicates_summary %>%
  dplyr::filter(mean_conc_soln > ASQL) %>%
dplyr::select(element,
              symbol,
              sample,
              mean_c_sample,
              mean_rec,
              sd_rec,
              sigma_2_rec,
              mean_conc_soln,
              c_std,
              LOD,
              LOQ,
              ASDL,
              ASQL) %>%
dplyr::mutate(QC = "Fortified Sample")
) %>%
dplyr::filter(!is.na(mean_rec)) %>%
dplyr::mutate(range = if_else(
  mean_rec >= 80 & mean_rec <= 120,
  "inside",
  "outside"

```

```

    )) %>%
dplyr::group_by(symbol, range) %>%
dplyr::mutate(min_range = signif(min(mean_conc_soln/c_std), 2),
             max_range = signif(max(mean_conc_soln/c_std), 2)
             ) %>%
dplyr::ungroup()

```

### Code for data processing

```

range <-
  working_range %>%
  dplyr::select(symbol, range, min_range, max_range) %>%
  dplyr::filter(range == "inside") %>%
  dplyr::select(symbol, min_range, max_range) %>%
  dplyr::distinct()

dilutions %>%
  dplyr::select(symbol, c_std) %>%
  dplyr::filter(symbol %in% data_out$symbol) %>%
  dplyr::distinct() %>%
  dplyr::slice_head(n = 10) %>% # duplicates each entry so remove the first set of values
  dplyr::right_join(range) %>%
  dplyr::mutate(min_soln_conc = signif(c_std * min_range, 2),
              max_conc_soln = signif(c_std * max_range, 2),
              min_samp_conc = signif(c_std * min_range*200, 2),
              max_samp_conc = signif(c_std * max_range*200, 2),
              c_std = signif(c_std, 2)
              ) %>%
  dplyr::rename(min_ratio = min_range,
              max_ratio = max_range) %>%
  gt()

```

### Code for data processing

```

ref_mat_summary %>%
  dplyr::filter(element %in% single_line) %>%
  dplyr::mutate(across(-c("symbol", "certificate_value", "plus_minus", "sample", "element", "date",
"certified"), ~signif(., digits = 3))) %>%
  dplyr::mutate(FOM = if_else(
    mean_conc_soln > ASQL,
    "",
    if_else(
      mean_conc_soln <= ASQL & mean_conc_soln > ASDL,
      "Trace",
      "< LOD"
    )
  ),
  ),
  `RSD (%)` = signif(sd_conc/mean_conc*100, digits = 3) %>%
  dplyr::select(symbol, element, sample, certified, FOM, mean_conc, sd_conc, `RSD (%)`, c_std, n,
z_score, certificate_value, plus_minus, mean_conc_soln, sd_conc_soln, ASDL, ASQL, LOD, LOQ) %>%
  dplyr::arrange(symbol, sample) %>%
  dplyr::rename(`Wavelength (nm)` = element,
              Element = symbol,
              Certificate = certified,
              `Z Score` = z_score,
              `Standard Concentration (mg/kg)` = c_std,

```

```

`Certificate Value (mg/kg)` = certificate_value,
`Certificate Uncertainty (mg/kg)` = plus_minus,
`Reference Material` = sample,
`Figure of Merit Indicator` = FOM,
`Conc. Mean (mg/kg)` = mean_conc,
`Conc. Stdev. (mg/kg)` = sd_conc,
`Soln. Conc. Mean (mg/kg)` = mean_conc_soln,
`Soln. Conc. Stdev. (mg/kg)` = sd_conc_soln,
`Analytical Solution Detection Limit (ASDL)` = ASDL,
`Analytical Solution Quantitation Limit (ASQL)` = ASQL,
`Limit of Detection (LOD)` = LOD,
`Limit of Quantitation (LOQ)` = LOQ
)

```

### Code for data processing

```

FAP_replicates_summary %>%
  dplyr::mutate(across(-c("symbol", "sample", "element", "FAP"), ~signif(., digits = 3))) %>%
  dplyr::mutate(FOM = if_else(
    mean_conc_soln > ASQL,
    "",
    if_else(
      mean_conc_soln <= ASQL & mean_conc_soln > ASDL,
      "Trace",
      "< LOD"
    )
  ),
),
`RSD (%)` = signif(sd_rec/mean_rec*100, digits = 3) %>%
  dplyr::select(symbol, element, sample, FAP, FOM, mean_rec, sd_rec, `RSD (%)`, c_std, n, mean_conc,
sd_conc, mean_conc_soln, sd_conc_soln, mean_spike_soln, ASDL, ASQL, LOD, LOQ) %>%
  dplyr::arrange(symbol, sample, FAP) %>%
  dplyr::mutate(FAP = str_replace(FAP, "FAP1", "Level 1"),
    FAP = str_replace(FAP, "FAP2", "Level 2"),
    FAP = str_replace(FAP, "FAP3", "Level 3"),
    FAP = factor(FAP, levels = c("Level 1", "Level 2", "Level 3")),
    sample = str_replace(sample, "breaded-chicken", "Breaded Chicken"),
    sample = str_replace(sample, "corn-flakes", "Corn Flakes"),
    sample = str_replace(sample, "italian-dressing", "Italian Dressing"),
    sample = str_replace(sample, "veg-oil", "Vegetable Oil")) %>%
  dplyr::rename(`Wavelength (nm)` = element,
    Element = symbol,
    Sample = sample,
    `Standard Concentration (mg/kg)` = c_std,
    `Fortified Analytical Portion (FAP) Level` = FAP,
    `Figure of Merit Indicator` = FOM,
    `Recovery Mean (%)` = mean_rec,
    `Recovery Stdev. (%)` = sd_rec,
    `Native Conc. Mean (mg/kg)` = mean_conc,
    `Native Conc. Stdev. (mg/kg)` = sd_conc,
    `Soln. Conc. Mean (mg/kg)` = mean_conc_soln,
    `Soln. Conc. Stdev. (mg/kg)` = sd_conc_soln,
    `Spike Conc. (mg/kg)` = mean_spike_soln,
    `Analytical Solution Detection Limit (ASDL)` = ASDL,
    `Analytical Solution Quantitation Limit (ASQL)` = ASQL,
    `Limit of Detection (LOD)` = LOD,

```

```
`Limit of Quantitation (LOQ)` = LOQ
)
```

Time saved

```
analysis_time <-
  data_out %>%
  dplyr::mutate(date_time = mdy_hms(date_time)) %>%
  dplyr::arrange(date_time) %>%
  dplyr::distinct(label, date_time)
```

```
analysis_time
```

Total number of solutions analyzed in one batch

```
nrow(analysis_time)
```

The total time in hours to run 120 solutions

```
time_length(
  interval(
    analysis_time %>%
      dplyr::slice(1) %>%
      dplyr::pull(date_time),
    analysis_time %>%
      dplyr::slice(120) %>%
      dplyr::pull(date_time)
  ),
  unit = "hour"
)
```

And what is the time in minutes between two successive solutions

```
time_length(
  interval(analysis_time %>%
    dplyr::slice(1) %>%
    dplyr::pull(date_time),
    analysis_time %>%
    dplyr::slice(2) %>%
    dplyr::pull(date_time)
  ),
  unit = "minute"
)
```

```
time_length(
  interval(analysis_time %>%
    dplyr::slice(59) %>%
    dplyr::pull(date_time),
    analysis_time %>%
    dplyr::slice(60) %>%
    dplyr::pull(date_time)
  ),
  unit = "minute"
)
```

```
time_length(
  interval(analysis_time %>%
```

```
dplyr::slice(119) %>%  
dplyr::pull(date_time),  
analysis_time %>%  
dplyr::slice(120) %>%  
dplyr::pull(date_time)  
)  
unit = "minute"  
)
```

## References

1. J. T. Sloop, H. J. B. Bonilla, T. Harville, B. T. Jones and G. L. Donati, *Talanta*, 2019, **205**, 120160.

## Packages and system information

```
sessionInfo()
```