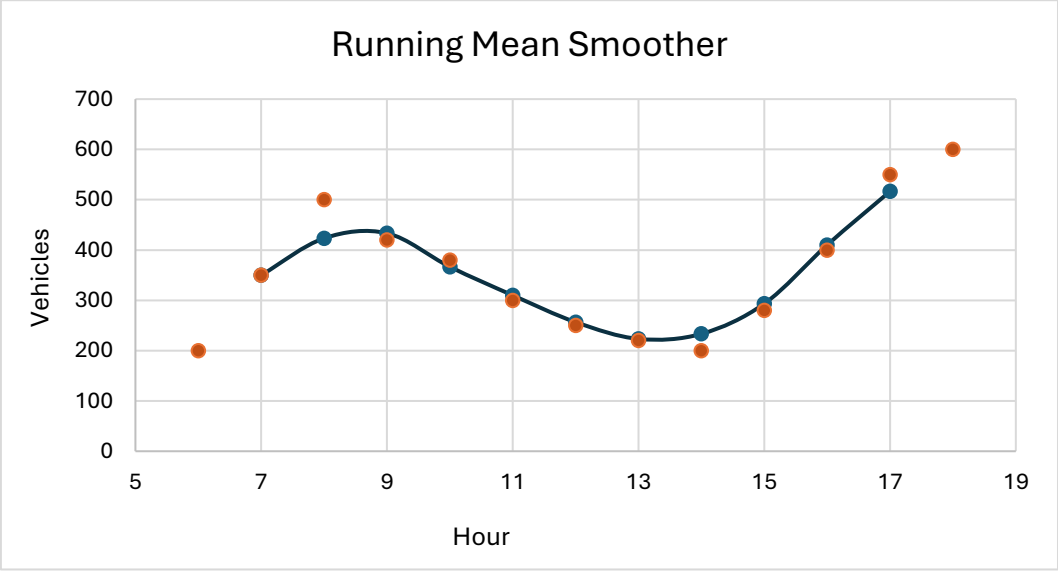
PART 2: DATA SMOOTHING

(a-1) Compute a running mean smoother by hand

Running Mean Smoother			
Window width		3 hours	
Time	Hour	Ave.Veh	Ave.Veh
06:00	6	-	N/A
07:00	7	(200+350+500)/3	350.000
08:00	8	(350+500+420)/3	423.333
09:00	9	(500+420+380)/3	433.333
10:00	10	(420+380+300)/3	366.667
11:00	11	(380+300+250)/3	310.000
12:00	12	(300+250+220)/3	256.667
13:00	13	(250+220+200)/3	223.333
14:00	14	(220+200+280)/3	233.333
15:00	15	(200+280+400)/3	293.333
16:00	16	(280+400+550)/3	410.000
17:00	17	(400+550+600)/3	516.667
18:00	18	-	N/A

(a-2) Draw your solution on a scatter plot of the data



(a-3) Validate your solution in R

------ (Code and Output shown in the following section) -----

(b) Use R to compute a running mean smoother using ksmooth()

------ (Code and Output shown in the following section) -----

(c-1) Create a Gaussian kernel smoother in Excel

σ	2
λ = 2*σ^2	8
Weights	wi

Gaussian Kernel Smoother in Excel			
Standard deviation is two hours			
Time	Hour	Vehicles	Gaussian Kernel Values
06:00	6	200	338.0665
07:00	7	350	360.0905
08:00	8	500	372.6262
09:00	9	420	369.2072
10:00	10	380	348.6222
11:00	11	300	318.1419
12:00	12	250	291.1620
13:00	13	220	281.2593
14:00	14	200	296.4157
15:00	15	280	335.1786
16:00	16	400	387.3647
17:00	17	550	440.2893
18:00	18	600	485.3281

(c-2) Validate your solution by hand

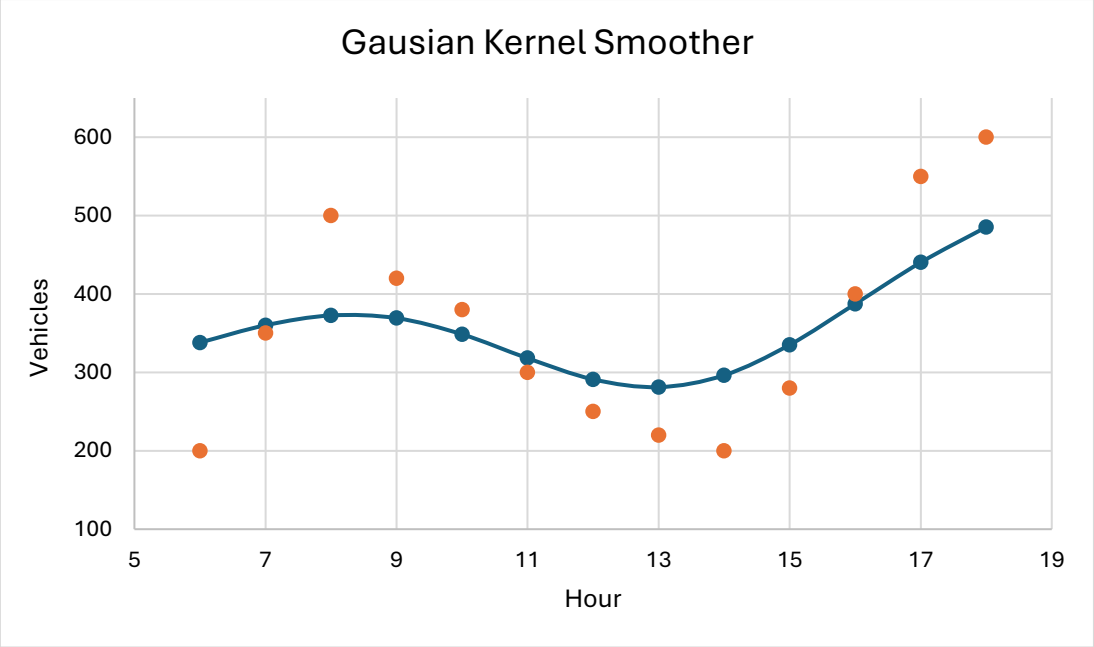
The estimator for f(·), denoted as $\hat{f}_\lambda(\cdot)$, is defined as follows:

$$\hat{f}_\lambda(\cdot) = \frac{\sum_{i=1}^{n} w_i y_i}{\sum_{j=1}^{n} w_i}$$

The weights are defined as $w_i = exp\left(-\frac{(x-x_i)^2}{\lambda}\right)$, i.e. the window is infinitely wide, but distant observations obtain little weight.

			06:00		07:00		08:00		09:00		10:00	
Time	**Hour (xi)**	**Vehicles (y,)**	**wi (xi =6)**	**w*y**	**wi (xi =7)**	**w*y**	**wi (xi =8)**	**w*y**	**wi (xi =9)**	**w*y**	**wi (xi =10)**	**w*y**
)6:00	6	200	1.0000	200.0000	0.8825	176.4994	0.6065	121.3061	0.3247	64.9305	0.1353	27.0671
)7:00	7	350	0.8825	308.8739	1.0000	350.0000	0.8825	308.8739	0.6065	212.2857	0.3247	113.6284
)8:00	8	500	0.6065	303.2653	0.8825	441.2485	1.0000	500.0000	0.8825	441.2485	0.6065	303.2653
)9:00	9	420	0.3247	136.3540	0.6065	254.7429	0.8825	370.6487	1.0000	420.0000	0.8825	370.6487
.0:00	10	380	0.1353	51.4274	0.3247	123.3679	0.6065	230.4817	0.8825	335.3488	1.0000	380.0000
.1:00	11	300	0.0439	13.1811	0.1353	40.6006	0.3247	97.3957	0.6065	181.9592	0.8825	264.7491
.2:00	12	250	0.0111	2.7772	0.0439	10.9842	0.1353	33.8338	0.3247	81.1631	0.6065	151.6327
.3:00	13	220	0.0022	0.4812	0.0111	2.4440	0.0439	9.6661	0.1353	29.7738	0.3247	71.4235
.4:00	14	200	0.0003	0.0671	0.0022	0.4375	0.0111	2.2218	0.0439	8.7874	0.1353	27.0671
.5:00	15	280	0.0000	0.0112	0.0003	0.0939	0.0022	0.6125	0.0111	3.1105	0.0439	12.3023
.6:00	16	400	0.0000	0.0015	0.0000	0.0160	0.0003	0.1342	0.0022	0.8750	0.0111	4.4436
.7:00	17	550	0.0000	0.0001	0.0000	0.0020	0.0000	0.0220	0.0003	0.1845	0.0022	1.2031
.8:00	18	600	0.0000	0.0000	0.0000	0.0002	0.0000	0.0022	0.0000	0.0240	0.0003	0.2013
Sum			3.0066	1016.4402	3.8891	1400.4371	4.4957	1675.1988	4.8203	1779.6910	4.9556	1727.6321
Smoothed Value			**338.0665**		**360.0905**		**372.6262**		**369.2072**		**348.6222**	

(c-3) Draw your solution on a scatter plot of the data



(c-4) Validate your solution in R

------ (Code and Output shown in the following section) -----

(d) Create the Gaussian kernel smoother with R, using the function ksmooth()

------ (Code and Output shown in the following section) -----

(e) Use a LOESS smoother for this data set

------ (Code and Output shown in the following section) -----