Simulation of Procurement Model

Jiachuan Tian

March 14, 2016

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

This document concerns about the simulation of procurement model with search cost (Chapter 3 of my dissertation).

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.2.3
```

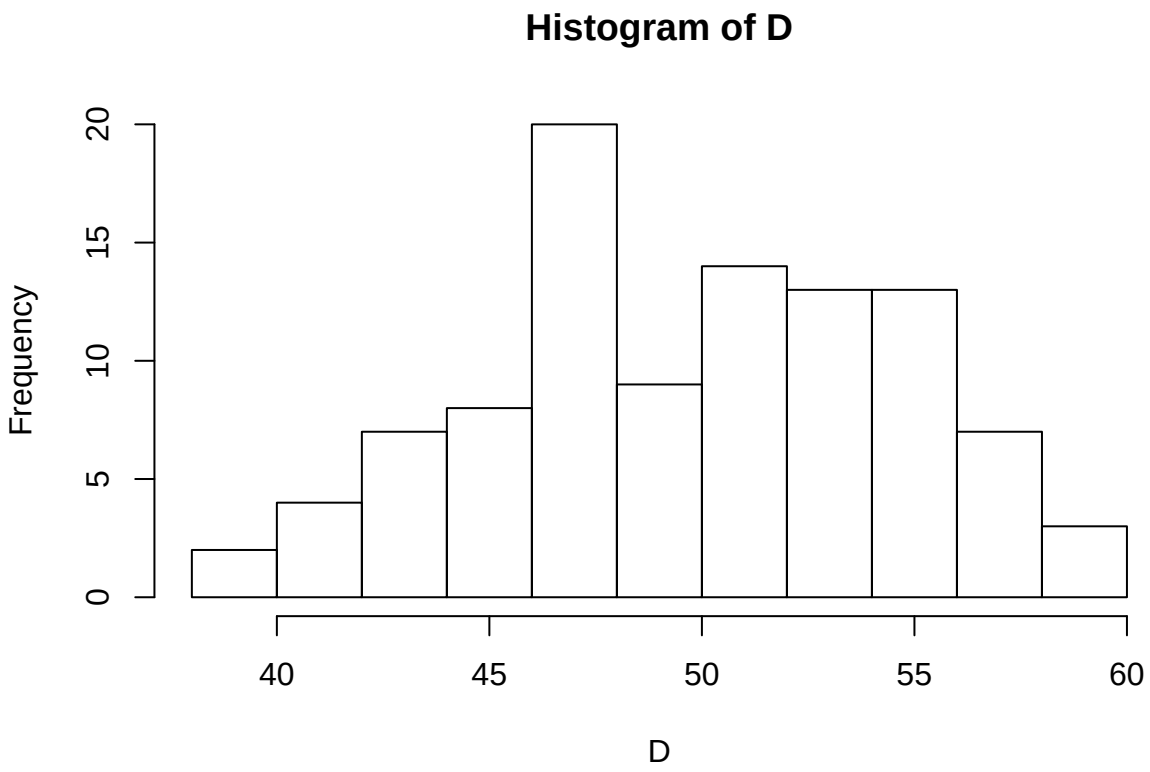
```
library(knitr)
```

```
## Warning: package 'knitr' was built under R version 3.2.3
```

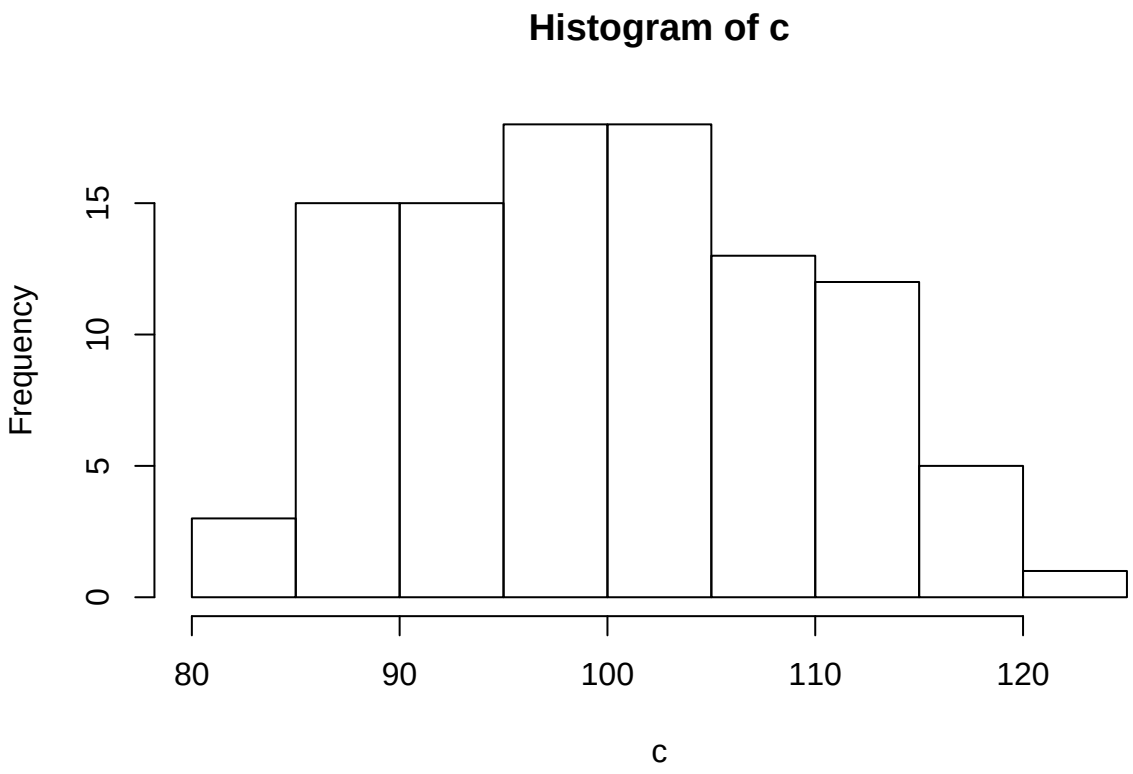
```
set.seed(1986)
```

Market Initialization

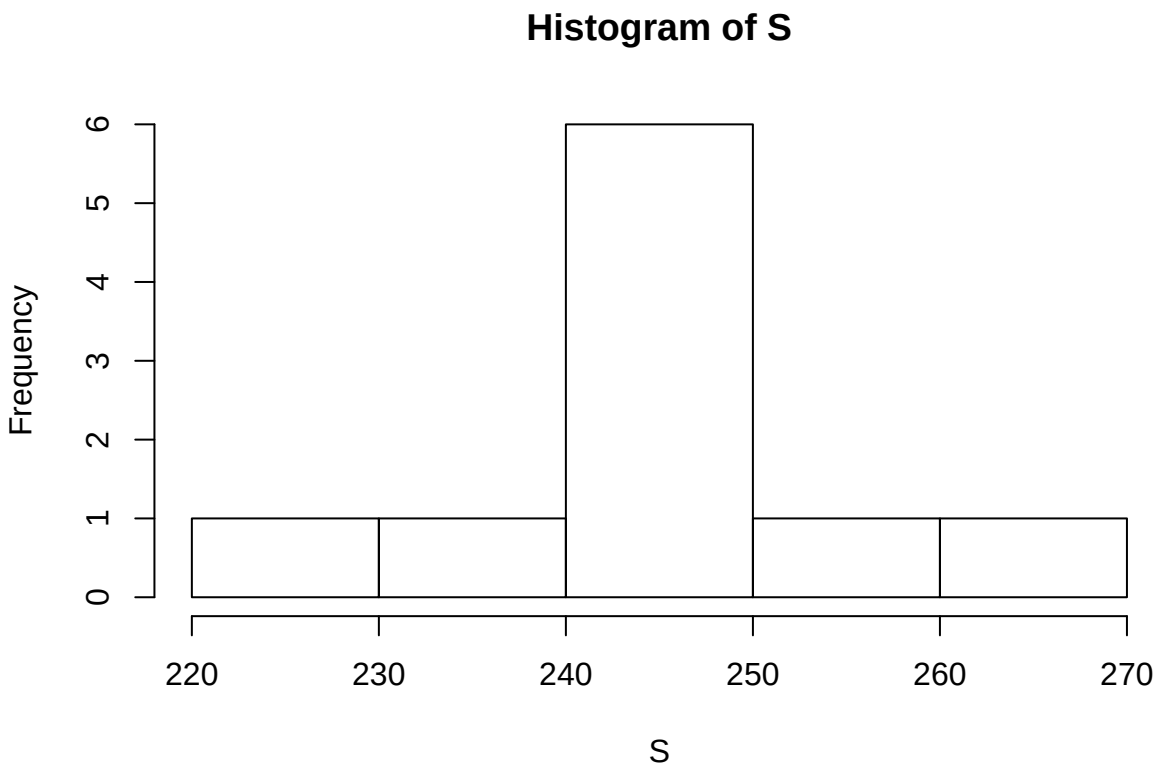
```
I = 100 #number of buyers
K = 10 #number of suppliers

D = rbinom(I, 100, 0.5); hist(D)
```




```
#alpha = rbeta(I, shape1 = 2, shape2 =2); hist(alpha)
c = rnorm(I, mean = 100, sd = 10); hist(c)
```




```
S = rbinom(K, 500, 0.5); hist(S)
```




```
Z = sum(D) - sum(S)
```

Buyer Side

```
WC_raw = rweibull(I*K, shape = 1, scale = 1)
WC = matrix(data = WC_raw, nrow = K, ncol = I) #each column corresponds to a buyer
event = rank(t(WC))

buyers_visit_order = sapply(1:I, function(i) {rank(WC[,i])}) #each column corresponds to a buyer

#
optimal_bid_fn = function(buyer_index){
  i = buyer_index
  order_local = buyers_visit_order[,i]
  Z_tilda_local = sapply(1:K, function(k){sum(D) - K * mean(S[order_local][1:k])})
  bid_local = (K-2)/(K-1) + Z_tilda_local^2 / 100000 + c[i]
  return(bid_local)
}
bid = sapply(1:I, function(i) optimal_bid_fn(i))

Z_tilda = sapply(1:I, function(i){sapply(1:K, function(k){sum(D) - K * mean(S[buyers_visit_order[,i]][1
search_cost = Z_tilda^2 / 100000 ########Attemtion!#
#should be the same as bid
bid_opt = (K-2)/(K-1) + t(c)%x%t(t(rep(1, K))) + search_cost
```

```
  #
index_loc_fn = function(buyer_index){
  return(sapply(1:K, function(k){which(buyers_visit_order[,buyer_index] == k)}))
}
increasing_index_loc = sapply(1:I, function(i) index_loc_fn(i))
```

Thus the buyers visit orders are:

```
buyers_visit_order_out = data.frame(t(buyers_visit_order))
name_K = lapply(1:K, function(i){ paste("Supplier", as.character(i))})
name_I = lapply(1:I, function(i)paste("Buyer", as.character(i)))
rownames(buyers_visit_order_out) = name_I
colnames(buyers_visit_order_out) = name_K
knitr::kable(buyers_visit_order_out, digits = 0, caption = "Buyers Visit Order ")
```

Table 1: Buyers Visit Order

	Supplier 1	Supplier 2	Supplier 3	Supplier 4	Supplier 5	Supplier 6	Supplier 7	Supplier 8	S
Buyer 1	10	6	4	2	3	1	8	5	
Buyer 2	4	10	3	7	2	1	9	8	
Buyer 3	8	10	1	5	6	3	2	9	
Buyer 4	4	10	9	6	2	1	5	7	
Buyer 5	1	5	4	10	6	8	2	7	
Buyer 6	7	9	6	3	8	2	10	5	
Buyer 7	10	5	7	4	9	2	1	8	
Buyer 8	6	5	2	1	9	4	7	3	
Buyer 9	7	1	3	8	9	6	4	2	
Buyer 10	9	6	2	8	7	4	5	10	
Buyer 11	1	10	7	3	8	9	2	5	
Buyer 12	4	5	1	9	3	10	6	8	
Buyer 13	7	8	4	3	2	6	5	10	
Buyer 14	6	4	7	3	9	5	1	8	
Buyer 15	6	7	5	8	2	3	10	9	
Buyer 16	1	9	6	5	10	4	7	3	
Buyer 17	6	9	1	8	4	3	7	5	
Buyer 18	1	3	8	9	7	6	10	5	
Buyer 19	2	3	4	8	6	7	9	10	
Buyer 20	1	5	4	2	8	6	7	9	
Buyer 21	6	5	3	8	2	1	9	4	
Buyer 22	8	7	5	3	6	4	10	2	
Buyer 23	7	9	5	1	6	2	8	3	
Buyer 24	3	4	5	8	7	6	10	9	
Buyer 25	5	6	3	7	8	1	9	10	
Buyer 26	6	1	7	2	9	4	5	10	
Buyer 27	2	4	10	6	9	8	7	3	
Buyer 28	4	10	5	3	7	2	6	8	
Buyer 29	10	6	9	7	5	4	2	1	
Buyer 30	10	1	3	9	4	7	6	2	
Buyer 31	1	10	8	4	6	3	5	2	
Buyer 32	9	6	7	1	10	8	2	5	
Buyer 33	10	3	1	2	8	9	5	4	
Buyer 34	2	7	5	6	9	3	4	8	

	Supplier 1	Supplier 2	Supplier 3	Supplier 4	Supplier 5	Supplier 6	Supplier 7	Supplier 8	S
Buyer 35	3	6	10	5	2	9	1	7	
Buyer 36	6	2	5	1	10	4	3	9	
Buyer 37	10	4	8	5	9	6	3	1	
Buyer 38	4	10	8	9	7	3	2	5	
Buyer 39	4	5	3	10	2	9	8	1	
Buyer 40	8	10	2	6	1	3	4	7	
Buyer 41	2	8	4	5	10	9	1	7	
Buyer 42	10	3	7	6	4	5	1	9	
Buyer 43	3	10	2	6	4	8	7	9	
Buyer 44	10	7	9	2	6	8	5	1	
Buyer 45	6	2	9	3	1	10	5	8	
Buyer 46	1	8	10	6	2	7	5	3	
Buyer 47	5	2	3	7	6	8	9	4	
Buyer 48	8	7	1	2	9	6	4	3	
Buyer 49	3	8	10	1	4	6	5	2	
Buyer 50	4	6	8	10	2	7	5	3	
Buyer 51	9	6	1	2	8	4	5	3	
Buyer 52	2	10	1	6	7	3	4	9	
Buyer 53	6	3	4	5	9	7	8	1	
Buyer 54	10	1	8	6	2	9	3	4	
Buyer 55	9	8	10	2	4	5	3	7	
Buyer 56	8	3	1	10	6	9	2	7	
Buyer 57	4	8	5	6	3	9	1	2	
Buyer 58	1	5	4	3	10	7	8	6	
Buyer 59	8	6	3	9	2	5	1	7	
Buyer 60	7	9	4	6	10	1	5	3	
Buyer 61	1	10	2	6	4	7	5	3	
Buyer 62	5	7	9	2	4	8	3	1	
Buyer 63	8	7	9	5	2	4	3	1	
Buyer 64	9	8	7	3	5	4	2	10	
Buyer 65	8	7	5	6	10	3	9	4	
Buyer 66	5	6	2	4	3	1	10	9	
Buyer 67	7	9	6	2	10	3	8	5	
Buyer 68	5	1	3	6	9	4	10	7	
Buyer 69	1	6	10	4	2	7	5	9	
Buyer 70	8	5	2	4	10	6	9	3	
Buyer 71	6	2	1	8	10	7	9	3	
Buyer 72	4	7	5	2	1	10	6	8	
Buyer 73	3	6	1	9	2	10	4	5	
Buyer 74	1	4	6	7	2	9	8	5	
Buyer 75	6	5	7	1	8	2	9	4	
Buyer 76	3	4	6	9	2	8	5	7	
Buyer 77	9	8	3	1	2	10	5	7	
Buyer 78	5	3	1	8	10	4	6	2	
Buyer 79	8	2	5	9	4	6	7	10	
Buyer 80	7	8	9	10	5	2	3	4	
Buyer 81	9	8	10	7	2	6	4	5	
Buyer 82	4	8	3	7	2	10	6	9	
Buyer 83	1	8	7	3	5	6	9	4	
Buyer 84	8	2	9	4	5	3	10	6	
Buyer 85	9	7	2	10	5	3	8	1	
Buyer 86	2	5	7	1	6	4	3	9	

	Supplier 1	Supplier 2	Supplier 3	Supplier 4	Supplier 5	Supplier 6	Supplier 7	Supplier 8	S
Buyer 87	3	8	9	2	10	4	7	1	
Buyer 88	7	6	4	5	1	8	9	3	
Buyer 89	3	7	2	5	8	1	9	4	
Buyer 90	7	8	10	2	3	1	6	9	
Buyer 91	1	5	10	2	8	4	6	7	
Buyer 92	3	2	5	6	10	7	9	4	
Buyer 93	7	9	6	2	1	8	5	4	
Buyer 94	2	3	6	1	4	8	5	9	
Buyer 95	2	9	4	1	6	10	3	8	
Buyer 96	10	8	7	9	4	5	6	3	
Buyer 97	2	7	5	4	3	10	9	1	
Buyer 98	3	8	4	7	2	1	9	6	
Buyer 99	2	6	9	4	5	8	1	7	
Buyer 100	3	4	5	9	8	2	6	1	
The table re	ads as follow	s:							
for example,	the first ro	w of the tabl	e is						

```
knitr::kable(buyers_visit_order_out[1,], digits = 0, caption = "Buyer 1 Visit Order ")
```

Table 2: Buyer 1 Visit Order

	Supplier 1	Supplier 2	Supplier 3	Supplier 4	Supplier 5	Supplier 6	Supplier 7	Supplier 8	Supplier 9
Buyer 1	10	6	4	2	3	1	8	5	9

The means buyer 1 first visit supplier 6, then supplier 4, . . . etc. The last supplier buyer 1 will visit is supplier 1 (he may not visit the supplier 1 though, depending on whether his procurement requirement is satisfied.) Buyes Optimal bid prices are:

```
seller_bid = t(sapply(1:I, function(i){return(bid[,i][buyers_visit_order[,i]])}))
seller_bid = data.frame(seller_bid)
rownames(seller_bid) = name_I
colnames(seller_bid) = name_K
knitr::kable(seller_bid, digits =3, caption = "Optimal Bid")
```

Table 3: Optimal Bid

	Supplier 1	Supplier 2	Supplier 3	Supplier 4	Supplier 5	Supplier 6	Supplier 7	Supplier 8	Supplier
Buyer 1	183.205	181.121	179.991	181.626	179.037	182.898	183.667	180.367	183.4
Buyer 2	161.727	164.940	161.270	163.036	160.113	155.756	164.350	164.633	163.0
Buyer 3	153.936	152.507	163.930	155.512	154.781	156.705	157.944	152.712	157.0
Buyer 4	158.490	162.077	160.865	159.909	157.250	152.893	159.139	160.969	158.2
Buyer 5	172.004	169.128	169.560	170.250	171.141	170.648	173.833	170.969	170.3
Buyer 6	159.716	160.428	159.827	157.197	160.342	157.533	159.365	159.981	157.5
Buyer 7	180.340	178.710	179.377	178.761	180.488	182.874	180.033	180.866	178.9
Buyer 8	155.014	156.666	157.491	153.417	155.043	157.233	154.930	156.975	156.2
Buyer 9	176.888	175.728	177.082	176.934	176.969	178.365	179.964	177.509	177.5
Buyer 10	176.434	175.384	173.950	176.235	176.271	177.780	177.108	176.286	174.4
Buyer 11	170.571	168.817	169.831	168.681	170.376	169.880	169.537	170.675	170.0

	Supplier 1	Supplier 2	Supplier 3	Supplier 4	Supplier 5	Supplier 6	Supplier 7	Supplier 8	Supplier
Buyer 12	178.945	178.894	170.832	180.051	179.199	180.017	179.030	180.286	178.7
Buyer 13	176.344	176.298	175.851	176.149	180.925	175.468	175.876	176.286	176.0
Buyer 14	155.897	154.061	156.427	153.235	157.931	154.459	155.059	157.974	151.8
Buyer 15	171.249	170.965	171.855	170.559	167.245	169.786	169.581	170.645	166.7
Buyer 16	181.095	179.376	178.101	179.752	179.342	179.932	178.016	178.355	178.0
Buyer 17	167.075	167.587	164.715	167.630	169.566	166.566	166.881	167.246	165.2
Buyer 18	181.946	184.039	181.428	180.227	180.838	180.998	180.193	181.739	182.5
Buyer 19	160.337	157.484	161.103	160.273	160.337	160.192	160.961	161.154	160.5
Buyer 20	164.341	164.238	161.898	166.171	162.793	163.479	163.086	162.622	161.7
Buyer 21	154.459	154.563	152.561	152.454	153.420	149.347	152.219	155.243	153.8
Buyer 22	154.009	154.055	156.062	158.901	154.116	157.316	153.996	158.636	165.4
Buyer 23	162.907	162.364	161.404	159.469	161.335	159.469	162.537	161.849	161.3
Buyer 24	153.027	156.224	155.578	154.679	154.917	154.978	154.859	154.665	149.5
Buyer 25	176.095	176.043	174.405	175.486	175.264	179.466	174.063	174.029	173.8
Buyer 26	148.246	147.828	149.340	150.103	149.453	149.849	149.645	150.666	149.6
Buyer 27	172.511	174.125	177.338	176.691	176.747	176.457	176.521	174.000	177.0
Buyer 28	163.366	164.819	163.290	163.324	163.276	159.992	163.493	164.831	155.6
Buyer 29	177.479	175.647	177.627	175.864	177.172	175.773	175.900	177.172	176.2
Buyer 30	167.552	167.245	167.586	166.961	167.117	165.288	165.384	168.271	165.5
Buyer 31	162.059	160.305	160.767	161.024	160.083	164.504	160.305	161.024	160.8
Buyer 32	156.399	156.313	156.128	153.504	155.335	156.700	152.999	154.315	153.1
Buyer 33	151.060	151.094	150.753	150.243	151.651	151.380	153.230	151.009	153.1
Buyer 34	168.157	167.469	168.921	168.751	169.263	170.292	169.305	169.049	168.9
Buyer 35	168.072	169.345	169.566	169.669	167.482	168.354	168.241	169.625	169.7
Buyer 36	152.494	151.902	154.200	150.643	153.481	154.458	154.200	153.686	152.3
Buyer 37	154.600	156.224	154.869	155.525	154.634	154.890	154.976	154.292	154.6
Buyer 38	172.624	172.804	172.945	172.610	172.205	173.180	167.977	172.292	172.1
Buyer 39	185.243	185.471	184.864	186.695	184.611	186.900	187.286	177.511	185.3
Buyer 40	171.014	171.449	176.886	172.770	182.872	174.944	173.332	171.215	173.3
Buyer 41	173.039	168.129	169.226	168.838	167.602	167.922	167.295	168.469	167.9
Buyer 42	180.262	179.106	179.009	178.515	178.683	177.124	179.955	180.296	178.9
Buyer 43	161.599	162.246	161.429	161.684	161.047	161.365	161.502	161.655	160.9
Buyer 44	165.277	166.586	165.425	164.205	165.910	166.642	163.850	164.970	164.2
Buyer 45	165.046	164.285	166.927	164.200	163.025	165.863	164.944	167.293	165.9
Buyer 46	166.118	165.794	164.364	165.600	170.872	166.192	166.222	168.563	166.7
Buyer 47	152.109	154.848	153.549	153.475	153.981	152.134	152.120	152.775	152.3
Buyer 48	172.364	172.464	184.414	177.630	172.399	173.967	175.656	176.661	172.9
Buyer 49	162.706	160.023	159.561	158.235	162.357	159.339	159.868	164.466	160.1
Buyer 50	175.965	175.752	176.413	176.400	170.346	176.605	175.991	175.922	176.2
Buyer 51	163.239	162.897	161.373	160.868	163.474	162.387	164.751	162.217	163.7
Buyer 52	168.789	169.096	168.789	168.279	167.048	169.472	168.661	167.883	168.3
Buyer 53	158.801	156.303	158.716	158.767	160.835	160.640	160.877	157.962	158.7
Buyer 54	165.612	165.305	165.177	166.848	166.331	165.021	169.810	168.016	166.4
Buyer 55	168.028	167.850	167.709	172.348	169.852	167.504	170.679	168.134	165.8
Buyer 56	176.135	179.455	187.032	175.609	176.672	175.018	180.514	176.475	178.4
Buyer 57	164.393	163.685	163.929	163.536	165.690	163.621	154.231	164.135	163.4
Buyer 58	164.655	162.084	161.957	162.084	162.901	163.621	163.171	161.914	162.9
Buyer 59	173.938	174.369	174.976	172.736	176.813	173.628	184.126	174.382	173.9
Buyer 60	152.502	153.862	149.867	151.408	153.828	151.996	150.589	149.495	152.5
Buyer 61	174.586	172.833	173.552	170.665	172.398	171.725	170.597	173.209	171.6
Buyer 62	157.500	159.334	159.163	160.883	159.463	159.527	159.677	164.566	159.5
Buyer 63	182.343	182.150	181.979	184.427	186.584	185.134	184.393	193.368	182.4

	Supplier 1	Supplier 2	Supplier 3	Supplier 4	Supplier 5	Supplier 6	Supplier 7	Supplier 8	Supplier
Buyer 64	161.319	161.654	161.718	163.960	163.787	163.007	166.152	161.513	161.8
Buyer 65	153.065	154.406	155.518	154.912	153.245	158.149	153.279	156.172	164.6
Buyer 66	167.649	168.224	170.616	167.548	170.099	174.817	169.380	168.167	168.3
Buyer 67	168.683	169.394	166.921	166.499	168.331	166.163	169.308	166.904	166.4
Buyer 68	156.529	161.761	158.251	154.914	156.358	157.043	156.324	155.071	154.9
Buyer 69	169.439	165.517	167.685	164.973	167.123	166.577	165.450	166.473	167.2
Buyer 70	184.589	185.812	192.976	186.200	184.576	185.124	184.781	190.013	196.0
Buyer 71	164.626	161.812	160.553	163.917	163.391	164.184	164.454	162.913	165.2
Buyer 72	155.868	156.338	156.527	151.882	148.262	157.446	156.629	157.844	155.4
Buyer 73	152.053	152.053	151.545	153.075	150.786	152.870	151.799	151.951	151.7
Buyer 74	169.489	164.647	165.315	167.209	163.899	167.883	167.940	165.198	167.7
Buyer 75	166.987	167.491	166.405	161.962	165.005	166.035	164.948	165.391	164.8
Buyer 76	167.564	168.225	171.244	172.095	166.742	171.882	168.923	172.120	170.7
Buyer 77	161.382	162.622	163.684	159.231	165.702	161.062	162.920	163.113	163.4
Buyer 78	184.708	183.314	186.824	181.849	181.387	185.630	182.537	183.401	181.9
Buyer 79	161.043	166.532	162.020	161.243	164.678	161.385	161.080	161.095	172.5
Buyer 80	175.291	173.901	174.079	173.760	176.242	178.399	176.208	175.514	171.9
Buyer 81	179.906	180.049	180.100	179.282	184.739	180.733	181.853	181.439	183.0
Buyer 82	170.087	169.641	170.556	169.778	171.241	170.522	170.130	169.931	161.3
Buyer 83	160.556	158.879	160.260	162.473	161.808	160.816	158.836	161.207	158.8
Buyer 84	180.258	185.682	180.450	180.066	181.170	183.215	180.245	180.792	191.6
Buyer 85	185.684	186.152	182.789	184.620	184.826	183.295	186.180	182.789	184.5
Buyer 86	166.013	163.889	162.225	163.171	162.238	164.714	164.542	163.513	162.1
Buyer 87	181.093	178.164	178.791	183.376	178.471	180.355	177.872	177.146	179.9
Buyer 88	176.629	176.984	174.141	174.924	174.519	176.428	176.385	171.688	174.0
Buyer 89	170.712	173.696	170.290	174.455	172.266	170.543	171.903	172.330	174.3
Buyer 90	148.296	147.955	147.429	152.069	150.400	145.597	148.923	146.838	149.5
Buyer 91	185.282	187.059	183.528	187.112	183.926	185.152	184.850	184.248	185.8
Buyer 92	169.185	167.132	168.052	167.728	167.950	167.497	166.737	168.155	166.6
Buyer 93	157.525	156.068	156.897	154.088	154.088	156.125	154.899	154.214	155.9
Buyer 94	154.861	154.183	155.456	155.371	154.988	155.947	153.241	155.883	155.6
Buyer 95	158.570	159.051	157.180	159.335	157.474	159.642	155.806	159.207	157.5
Buyer 96	165.597	165.418	165.583	165.632	167.351	165.086	166.059	168.568	165.2
Buyer 97	163.976	163.650	163.520	163.469	166.111	165.048	163.835	164.741	163.7
Buyer 98	169.648	168.988	169.179	169.307	174.519	168.289	169.023	169.563	169.6
Buyer 99	159.546	161.415	161.273	157.537	159.293	161.330	160.818	161.623	161.1
Buyer 100	162.953	162.953	165.504	164.990	165.054	159.466	165.332	163.459	164.7

Seller Side

In this section we investigate sellers' problems.

```
seller_wc = t(sapply(1:I, function(i){return(WC[,i][buyers_visit_order[,i]])}))

#Case 1
#seller_wc = 10 * seller_wc

seller_value_order = sapply(1:K, function(k) rank(- (seller_bid[,k] - seller_wc[,k])))
```

For each supplier, based on the bid each buyer offers and the time they arrive, the supplier ranks them by bid price adjusted by waiting cost and chooses highest ones.

```
seller_value_order = data.frame(seller_value_order)
rownames(seller_value_order) = name_I
colnames(seller_value_order) = name_K
knitr::kable(seller_value_order, digits = 0 , caption = "Rank of Value of bids")
```

Table 4: Rank of Value of bids

	Supplier 1	Supplier 2	Supplier 3	Supplier 4	Supplier 5	Supplier 6	Supplier 7	Supplier 8	Supplier
Buyer 1	6	9	12	7	13	8	5	9	
Buyer 2	71	54	67	73	81	87	64	62	
Buyer 3	91	95	59	83	84	80	80	92	
Buyer 4	78	72	69	70	88	93	75	76	
Buyer 5	28	34	40	33	32	36	28	31	
Buyer 6	80	73	74	81	76	85	76	79	
Buyer 7	11	22	13	13	10	7	26	12	
Buyer 8	86	88	83	87	95	78	87	84	
Buyer 9	18	19	18	17	25	17	12	14	
Buyer 10	17	21	27	32	23	22	22	20	
Buyer 11	34	33	34	39	34	37	42	38	
Buyer 12	16	12	31	11	11	13	21	10	
Buyer 13	20	26	21	21	15	27	23	19	
Buyer 14	85	89	85	88	78	92	85	83	
Buyer 15	31	38	29	37	48	39	33	34	
Buyer 16	9	13	16	15	12	14	15	13	
Buyer 17	46	40	53	42	40	45	46	49	
Buyer 18	8	7	8	9	14	9	11	5	
Buyer 19	73	80	75	80	75	75	71	72	
Buyer 20	57	62	64	46	72	60	67	63	
Buyer 21	88	87	95	89	90	99	94	86	
Buyer 22	90	93	88	75	87	79	93	80	
Buyer 23	66	69	68	74	68	76	65	71	
Buyer 24	92	84	90	90	93	90	83	87	
Buyer 25	22	16	24	24	26	15	27	25	
Buyer 26	99	99	98	98	98	98	100	98	1
Buyer 27	27	24	17	20	19	20	20	27	
Buyer 28	59	58	61	59	61	86	59	56	
Buyer 29	15	18	19	19	17	23	24	22	
Buyer 30	49	52	45	50	54	55	56	42	
Buyer 31	65	76	70	68	73	57	73	77	
Buyer 32	82	85	86	92	82	83	92	97	
Buyer 33	98	97	100	100	96	97	95	96	
Buyer 34	44	43	42	38	39	35	32	40	
Buyer 35	51	35	38	35	43	40	38	36	
Buyer 36	94	96	94	99	92	89	89	91	
Buyer 37	87	82	87	84	85	88	86	95	
Buyer 38	32	27	28	30	28	30	53	32	
Buyer 39	4	3	4	3	2	1	1	15	

	Supplier 1	Supplier 2	Supplier 3	Supplier 4	Supplier 5	Supplier 6	Supplier 7	Supplier 8	Supplier
Buyer 40	35	32	23	28	7	24	29	33	
Buyer 41	30	41	37	49	42	42	43	39	
Buyer 42	10	15	14	14	16	18	14	11	
Buyer 43	67	63	72	64	70	65	68	67	
Buyer 44	55	48	51	53	53	44	62	58	
Buyer 45	64	53	48	62	65	48	52	52	
Buyer 46	50	49	56	52	35	49	47	45	
Buyer 47	97	86	92	97	91	94	99	93	
Buyer 48	26	29	5	16	31	29	19	16	
Buyer 49	70	74	79	76	66	74	74	55	
Buyer 50	29	20	20	18	38	21	18	21	
Buyer 51	62	61	71	69	57	64	58	65	
Buyer 52	42	36	39	40	45	38	39	46	
Buyer 53	76	81	76	77	71	69	69	81	
Buyer 54	54	55	52	47	46	54	34	47	
Buyer 55	41	44	44	31	37	43	40	41	
Buyer 56	23	11	3	23	22	26	10	18	
Buyer 57	58	60	55	61	51	63	88	60	
Buyer 58	56	71	65	63	62	61	66	74	
Buyer 59	25	23	22	27	20	28	6	23	
Buyer 60	95	91	97	93	86	96	98	99	
Buyer 61	24	30	30	34	30	31	31	26	
Buyer 62	79	75	77	72	74	70	77	57	
Buyer 63	7	8	7	5	1	4	8	1	
Buyer 64	72	79	66	60	64	59	54	69	
Buyer 65	93	94	93	85	89	77	90	85	
Buyer 66	43	37	32	41	33	25	36	44	
Buyer 67	45	31	46	55	41	51	37	70	
Buyer 68	84	67	78	86	79	82	79	88	
Buyer 69	37	51	43	51	44	46	50	50	
Buyer 70	3	5	1	2	5	5	3	2	
Buyer 71	60	66	73	54	60	62	55	61	
Buyer 72	83	83	84	95	100	81	82	82	
Buyer 73	96	98	96	91	97	95	96	94	
Buyer 74	36	56	60	45	59	58	44	53	
Buyer 75	47	46	49	66	55	50	60	51	
Buyer 76	48	57	36	29	49	32	35	30	
Buyer 77	68	68	57	79	58	68	63	64	
Buyer 78	5	6	2	6	8	3	7	4	
Buyer 79	74	47	63	65	63	67	72	73	
Buyer 80	21	25	25	26	21	16	25	24	
Buyer 81	12	10	11	12	4	12	9	6	
Buyer 82	38	39	33	36	36	34	41	35	
Buyer 83	69	77	80	67	69	71	78	75	
Buyer 84	13	2	10	10	9	10	13	8	
Buyer 85	2	4	9	4	3	6	2	7	
Buyer 86	52	59	62	57	67	53	57	66	
Buyer 87	14	14	15	8	18	11	17	17	
Buyer 88	19	17	26	22	24	19	16	29	
Buyer 89	33	28	35	25	29	33	30	28	
Buyer 90	100	100	99	96	99	100	97	100	
Buyer 91	1	1	6	1	6	2	4	3	

	Supplier 1	Supplier 2	Supplier 3	Supplier 4	Supplier 5	Supplier 6	Supplier 7	Supplier 8	Supplier
Buyer 92	40	45	47	44	47	47	45	48	
Buyer 93	81	92	89	94	94	84	84	89	
Buyer 94	89	90	91	82	83	91	91	90	
Buyer 95	77	78	82	71	80	72	81	78	
Buyer 96	53	50	50	48	56	52	49	43	
Buyer 97	61	65	58	58	50	56	61	54	
Buyer 98	39	42	41	43	27	41	48	37	
Buyer 99	75	70	81	78	77	66	70	68	
Buyer 100	63	64	54	56	52	73	51	59	

We solve for the case of single unit auction, we have

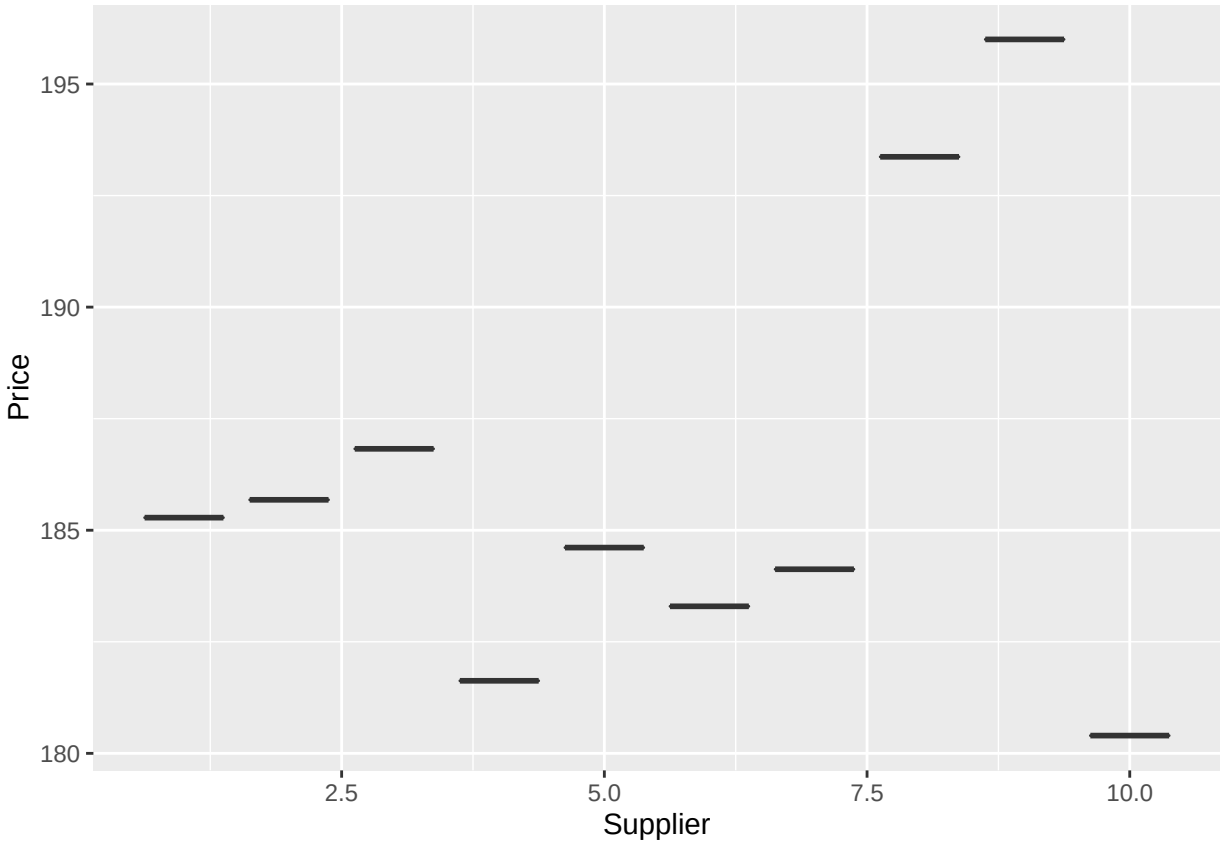
```
auction_winner = data.frame(c(91,84, 78, 1,39, 85, 59, 63, 70, 18))
auction_winner = t(auction_winner)
rownames(auction_winner) = NULL
colnames(auction_winner) = name_K
transaction_price = sapply(1:K, function(x) seller_bid[auction_winner[,x], x])
transaction_price_table = data.frame(t(transaction_price))
colnames(transaction_price_table) = name_K
knitr::kable(transaction_price_table, digits = 3 , caption = "Transaction Price")
```

Table 5: Transaction Price

Supplier 1	Supplier 2	Supplier 3	Supplier 4	Supplier 5	Supplier 6	Supplier 7	Supplier 8	Supplier 9	Supplier
185.282	185.682	186.824	181.626	184.611	183.295	184.126	193.368	196	180.

Plot the result

```
transaction_price_plot = data.frame(cbind(transaction_price,c(1:K)))[,c(2,1)]
colnames(transaction_price_plot) = c("Supplier", "Price")
ggplot(transaction_price_plot, aes(x = Supplier, y = Price, group = Supplier))+geom_boxplot()
```


195
190
185
180
Price
2.5
5.0
7.5
10.0
Supplier