

Program 2 Graph Analysis

Ryan Schaefer and Wes Anderson

Create Dataset

```
library(ggplot2)
library(ggpubr)
data = read.csv("WesDataRun3.csv")
data$n2 = data$size ^ 2
data$nlogn = log(data$size) * data$size
data
```

##	var_type	size	format	insertion_time	quick_time	merge_time
## 1	int	500000	noDuplicates	N/A	0.16627800	1.9073100
## 2	int	1000000	40duplicates	N/A	0.30342100	3.9111900
## 3	int	100000	40duplicates	N/A	0.02762030	0.4068040
## 4	int	10000	40duplicates	N/A	0.00277897	0.0383910
## 5	int	50000	sorted	N/A	0.00911402	0.1914150
## 6	int	50000	20duplicates	N/A	0.01518370	0.2008030
## 7	int	5000	noDuplicates	N/A	0.00121367	0.0178258
## 8	int	500000	sorted	N/A	0.11851300	1.9419900
## 9	int	500000	60sorted	N/A	0.14154100	2.0029200
## 10	int	10000	60sorted	N/A	0.00204434	0.0363390
## 11	int	1000000	noDuplicates	N/A	0.34930000	4.1095600
## 12	int	1000000	20duplicates	N/A	0.32045700	4.2411400
## 13	int	50000	noDuplicates	N/A	0.01441990	0.2063230
## 14	int	5000	60sorted	N/A	0.00114934	0.0190661
## 15	int	5000	sorted	N/A	0.00100552	0.0187336
## 16	int	100000	20duplicates	N/A	0.03032380	0.4040670
## 17	int	50000	60sorted	N/A	0.01140880	0.1997040
## 18	int	10000	noDuplicates	N/A	0.00266048	0.0395766
## 19	int	500000	20duplicates	N/A	0.15689000	2.1020900
## 20	int	500000	40duplicates	N/A	0.15787400	2.1274400
## 21	int	1000000	sorted	N/A	0.22663900	4.1182700
## 22	int	5000	20duplicates	N/A	0.00135010	0.0194802
## 23	int	100000	noDuplicates	N/A	0.03088480	0.4195140
## 24	int	50000	40duplicates	N/A	0.01508430	0.2063080
## 25	int	10000	20duplicates	N/A	0.00283506	0.0391143
## 26	int	100000	sorted	N/A	0.02221500	0.4058590
## 27	int	100000	60sorted	N/A	0.02243270	0.4056630
## 28	int	10000	sorted	N/A	0.00178320	0.0382798
## 29	int	5000	40duplicates	N/A	0.00140764	0.0192427
## 30	int	1000000	60sorted	N/A	0.25955200	4.2715500
## 31	string	50000	sorted	N/A	0.08538960	0.2713100
## 32	string	500000	20duplicates	N/A	0.99799200	3.3825700
## 33	string	50000	20duplicates	N/A	0.08378530	0.3125810

## 34	string	10000	40duplicates	N/A	0.01490460	0.0622808
## 35	string	10000	60sorted	N/A	0.01405230	0.0558130
## 36	string	100000	sorted	N/A	0.16371400	0.5824980
## 37	string	5000	40duplicates	N/A	0.00670211	0.0281871
## 38	string	500000	60sorted	N/A	1.01538000	3.1147400
## 39	string	50000	noDuplicates	N/A	0.08224720	0.3117670
## 40	string	500000	40duplicates	N/A	1.06405000	3.4492800
## 41	string	5000	20duplicates	N/A	0.00683657	0.0287495
## 42	string	100000	noDuplicates	N/A	0.17792100	0.6446040
## 43	string	5000	noDuplicates	N/A	0.00715408	0.0284622
## 44	string	100000	60sorted	N/A	0.18996400	0.6162350
## 45	string	1000000	20duplicates	N/A	2.27596000	7.0983900
## 46	string	10000	noDuplicates	N/A	0.02476040	0.0600707
## 47	string	1000000	noDuplicates	N/A	2.17305000	7.1805700
## 48	string	1000000	sorted	N/A	2.09597000	6.2435100
## 49	string	500000	noDuplicates	N/A	1.05809000	3.5065900
## 50	string	100000	40duplicates	N/A	0.19354900	0.6515980
## 51	string	5000	60sorted	N/A	0.00628840	0.0284520
## 52	string	1000000	60sorted	N/A	2.32343000	6.6549600
## 53	string	5000	sorted	N/A	0.00566607	0.0262629
## 54	string	100000	20duplicates	N/A	0.17994900	0.6602710
## 55	string	10000	20duplicates	N/A	0.02783630	0.0720578
## 56	string	10000	sorted	N/A	0.01428690	0.0547817
## 57	string	500000	sorted	N/A	1.00368000	3.0378100
## 58	string	50000	40duplicates	N/A	0.08948610	0.3212370
## 59	string	50000	60sorted	N/A	0.07936610	0.2995070
## 60	string	1000000	40duplicates	N/A	2.20410000	7.1878100
##	shell_time	intro_time	tim_time	n2	nlogn	
## 1	0.236014000	0.61839700	0.65286600	2.5e+11	6561181.69	
## 2	0.654883000	1.44884000	1.44927000	1.0e+12	13815510.56	
## 3	0.041638600	0.12844800	0.12899100	1.0e+10	1151292.55	
## 4	0.002489040	0.00938487	0.00983652	1.0e+08	92103.40	
## 5	0.004850850	0.05602080	0.05082480	2.5e+09	540988.91	
## 6	0.018829200	0.05359550	0.05601230	2.5e+09	540988.91	
## 7	0.001282920	0.00435888	0.00444830	2.5e+07	42585.97	
## 8	0.063586300	0.65536500	0.59006400	2.5e+11	6561181.69	
## 9	0.125113000	0.64147000	0.63442800	2.5e+11	6561181.69	
## 10	0.001436780	0.00908815	0.00884554	1.0e+08	92103.40	
## 11	0.549779000	1.41396000	1.48915000	1.0e+12	13815510.56	
## 12	0.590384000	1.43708000	1.51672000	1.0e+12	13815510.56	
## 13	0.019451000	0.05897490	0.06110110	2.5e+09	540988.91	
## 14	0.000662599	0.00481884	0.00445120	2.5e+07	42585.97	
## 15	0.000375610	0.00505910	0.00418441	2.5e+07	42585.97	
## 16	0.041764800	0.12128300	0.12764800	1.0e+10	1151292.55	
## 17	0.009759100	0.05681090	0.04973800	2.5e+09	540988.91	
## 18	0.002997810	0.00961265	0.00973909	1.0e+08	92103.40	
## 19	0.266548000	0.70128000	0.73711000	2.5e+11	6561181.69	
## 20	0.265188000	0.66922000	0.71977100	2.5e+11	6561181.69	
## 21	0.130444000	1.32907000	1.29050000	1.0e+12	13815510.56	
## 22	0.001214040	0.00475111	0.00501881	2.5e+07	42585.97	
## 23	0.041319800	0.12744900	0.12976900	1.0e+10	1151292.55	
## 24	0.018773200	0.05945190	0.06041140	2.5e+09	540988.91	
## 25	0.002809160	0.01008820	0.01080560	1.0e+08	92103.40	
## 26	0.010245500	0.10905600	0.10560100	1.0e+10	1151292.55	

```
## 27 0.021581800 0.11100500 0.11553100 1.0e+10 1151292.55
## 28 0.000856825 0.00846338 0.00888111 1.0e+08 92103.40
## 29 0.001237620 0.00475767 0.00517817 2.5e+07 42585.97
## 30 0.277285000 1.35969000 1.42055000 1.0e+12 13815510.56
## 31 0.055100800 0.14140700 0.12559300 2.5e+09 540988.91
## 32 2.272680000 2.05675000 2.27412000 2.5e+11 6561181.69
## 33 0.158706000 0.16776200 0.18272900 2.5e+09 540988.91
## 34 0.023841500 0.02624370 0.03123970 1.0e+08 92103.40
## 35 0.021971100 0.02694850 0.02622200 1.0e+08 92103.40
## 36 0.133409000 0.31513600 0.27932500 1.0e+10 1151292.55
## 37 0.009830370 0.01395100 0.01467340 2.5e+07 42585.97
## 38 2.128530000 2.00509000 1.94504000 2.5e+11 6561181.69
## 39 0.164745000 0.16827400 0.18605600 2.5e+09 540988.91
## 40 2.582990000 2.09678000 2.32441000 2.5e+11 6561181.69
## 41 0.009843180 0.01551400 0.01471600 2.5e+07 42585.97
## 42 0.363742000 0.35173400 0.40616200 1.0e+10 1151292.55
## 43 0.010023400 0.01466180 0.01272380 2.5e+07 42585.97
## 44 0.348958000 0.34787100 0.33249800 1.0e+10 1151292.55
## 45 5.831080000 4.59541000 4.97487000 1.0e+12 13815510.56
## 46 0.024462700 0.03325060 0.02924960 1.0e+08 92103.40
## 47 5.670330000 4.60570000 4.97796000 1.0e+12 13815510.56
## 48 1.516350000 4.31915000 3.55680000 1.0e+12 13815510.56
## 49 2.343110000 2.10105000 2.38171000 2.5e+11 6561181.69
## 50 0.379177000 0.38728100 0.40734700 1.0e+10 1151292.55
## 51 0.009884860 0.01393670 0.01320130 2.5e+07 42585.97
## 52 4.896730000 4.62821000 4.24579000 1.0e+12 13815510.56
## 53 0.004569390 0.01455490 0.00973213 2.5e+07 42585.97
## 54 0.390118000 0.38299500 0.41871400 1.0e+10 1151292.55
## 55 0.021781900 0.02884380 0.03224410 1.0e+08 92103.40
## 56 0.010329000 0.02842000 0.02182600 1.0e+08 92103.40
## 57 0.707174000 2.02560000 1.71959000 2.5e+11 6561181.69
## 58 0.164750000 0.17359600 0.18528500 2.5e+09 540988.91
## 59 0.154817000 0.16579700 0.15465400 2.5e+09 540988.91
## 60 5.600070000 4.91904000 5.01714000 1.0e+12 13815510.56
```

Insertion Sort

```
# insertionTimes = aggregate(insertion_time ~ var_type + size + n2 + format, data = data, FUN = mean)
# insertionTimes2 = aggregate(insertion_time ~ var_type + size + n2, data = data, FUN = mean)
# ggplot(insertionTimes2, aes(x = size, y = insertion_time, color = var_type)) +
#   geom_line() +
#   labs(title = "Mean Insertion Sort Time By Data Set Size and Data Type", x = "n", y = "Insertion Sort Time")
#   guides(color = guide_legend(title = "Data Type"))
```

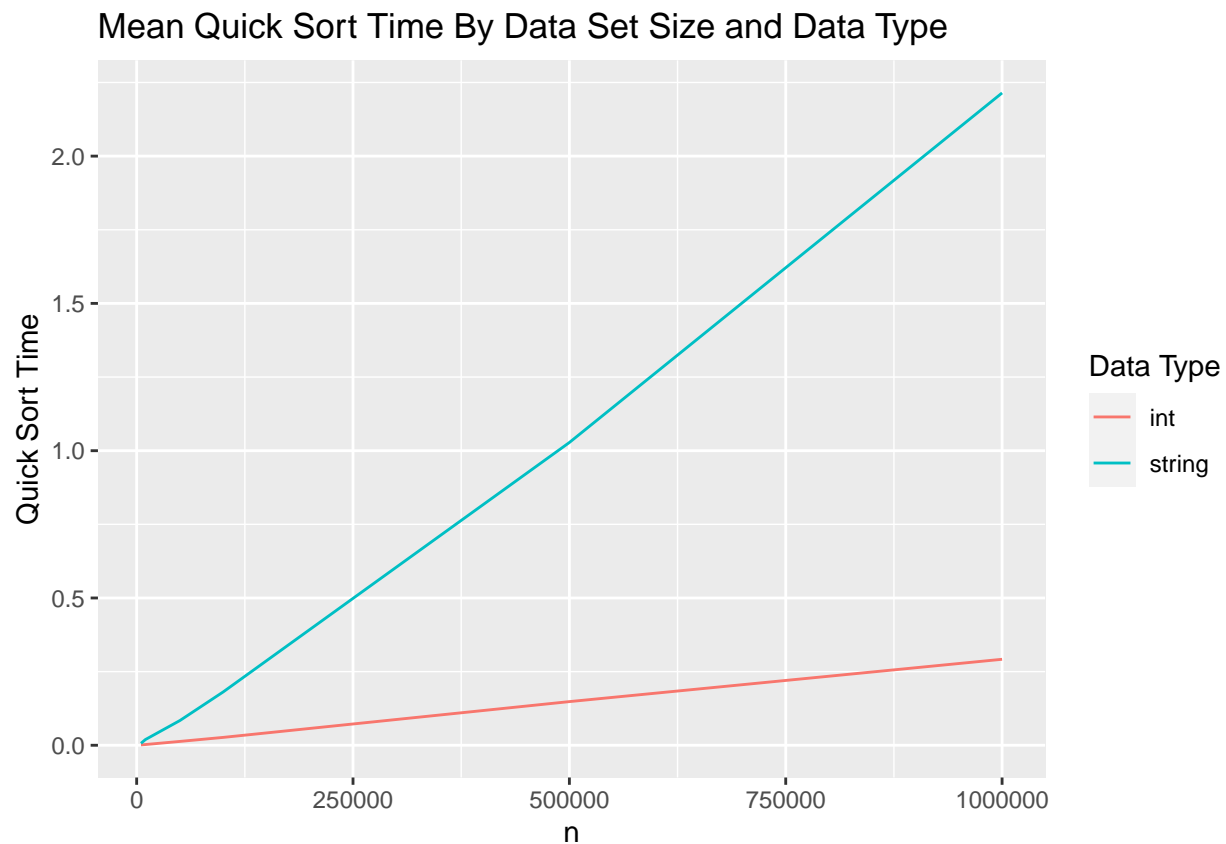
```
# ggplot(insertionTimes, aes(x = size, y = insertion_time, color = var_type)) +
#   labs(title = "Insertion Sort Regression Models By Data Type", x = "n^2", y = "Insertion Sort Time")
#   geom_smooth(method="lm") +
#   geom_point() +
#   stat_regline_equation(label.x=0, label.y=c(9000, 6000)) +
#   stat_cor(aes(label=..rr.label..), label.x=0, label.y=c(8000, 5000)) +
#   guides(color = guide_legend(title = "Data Type"))
```

```
# insertionInts = subset(insertionTimes, var_type == "int")
# ggplot(insertionInts, aes(x = size, y = insertion_time, color = format)) +
#   geom_line() +
#   labs(title = "Insertion Sort Time With Integer Data By Data Set Size and File Format", x = "n", y =
#   guides(color = guide_legend(title = "File Format"))
```

```
# insertionStrings = subset(insertionTimes, var_type == "string")
# ggplot(insertionStrings, aes(x = size, y = insertion_time, color = format)) +
#   geom_line() +
#   labs(title = "Insertion Sort Time With String Data By Data Set Size and File Format", x = "n", y =
#   guides(color = guide_legend(title = "File Format"))
```

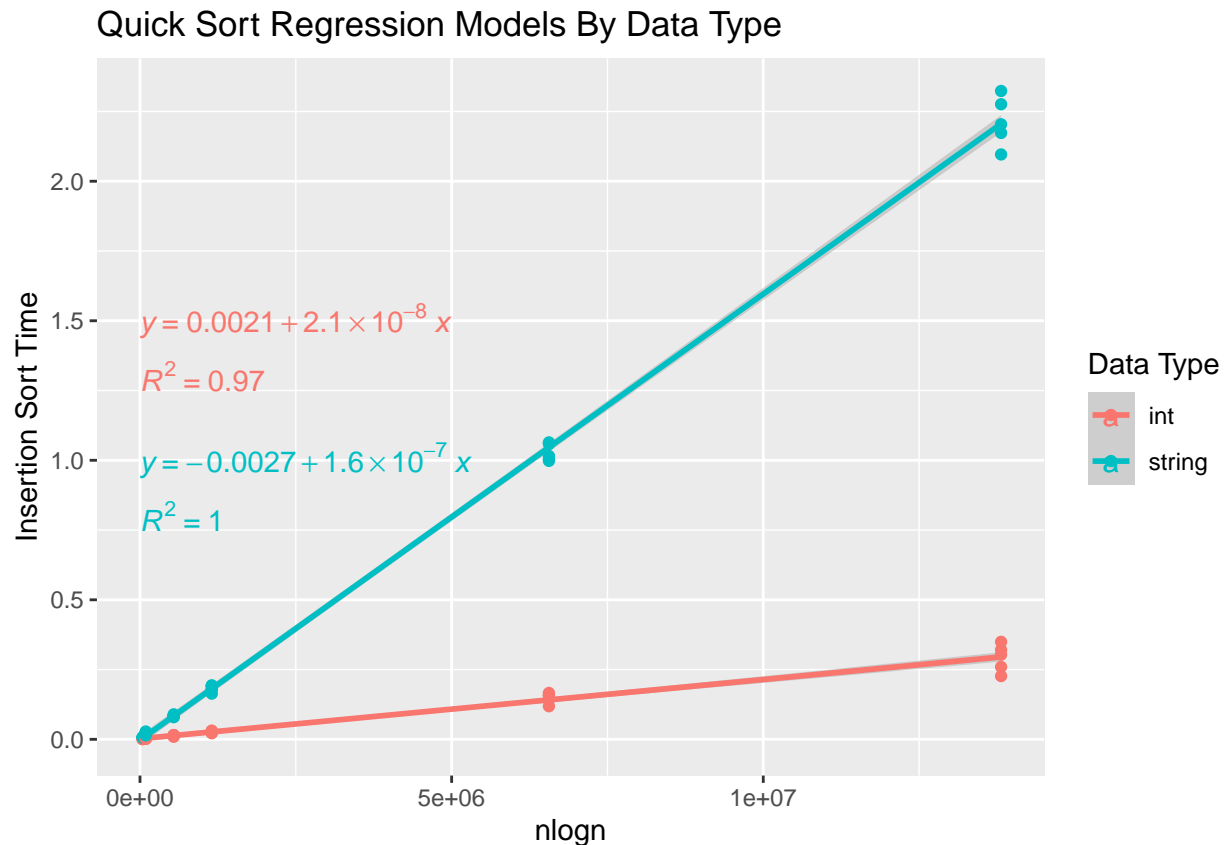
Quick Sort

```
quickTimes = aggregate(quick_time ~ var_type + size + nlogn + format, data = data, FUN = mean)
quickTimes2 = aggregate(quick_time ~ var_type + size + nlogn, data = data, FUN = mean)
ggplot(quickTimes2, aes(x = size, y = quick_time, color = var_type)) +
  geom_line() +
  labs(title = "Mean Quick Sort Time By Data Set Size and Data Type", x = "n", y = "Quick Sort Time") +
  guides(color = guide_legend(title = "Data Type"))
```



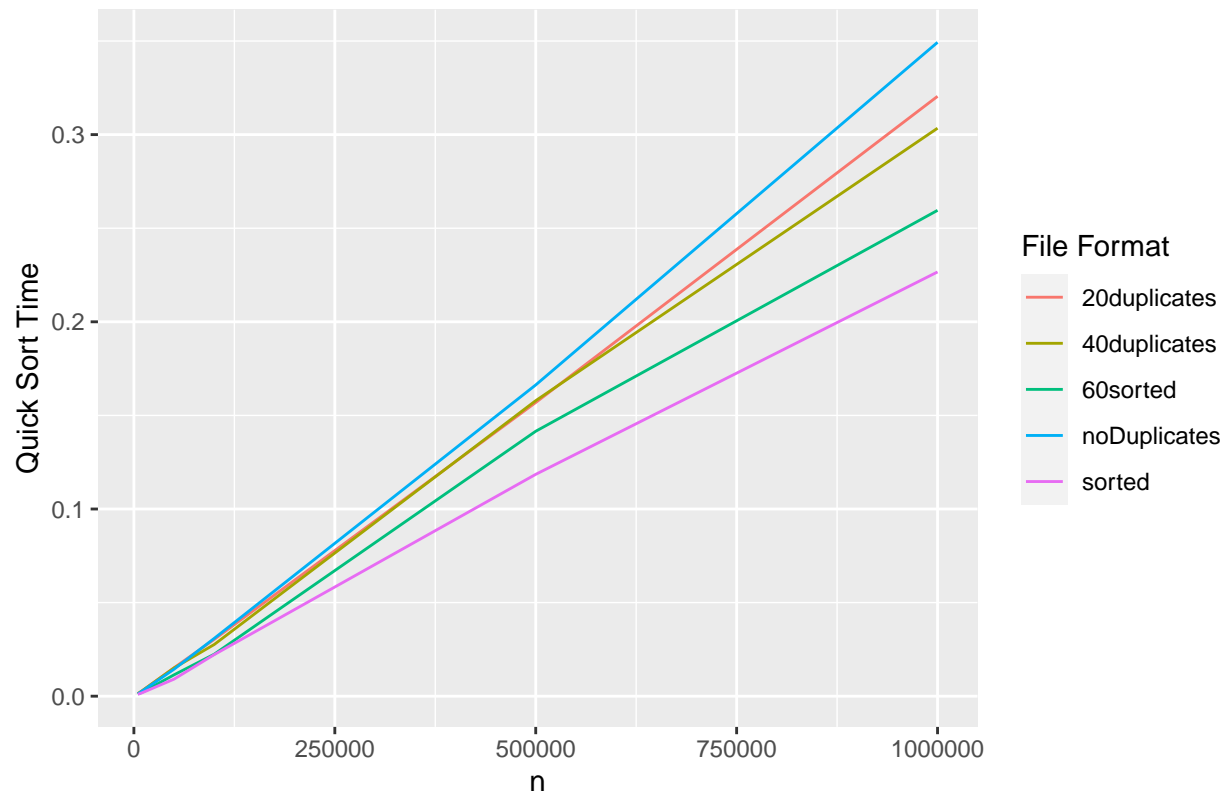
```
ggplot(quickTimes, aes(x = nlogn, y = quick_time, color = var_type)) +
  labs(title = "Quick Sort Regression Models By Data Type", x = "nlogn", y = "Insertion Sort Time") +
  geom_smooth(method="lm") +
  geom_point() +
  stat_regline_equation(label.x=0, label.y=c(1.5, 1)) +
  stat_cor(aes(label=..rr.label..), label.x=0, label.y=c(1.3, 0.8)) +
  guides(color = guide_legend(title = "Data Type"))
```

'geom_smooth()' using formula 'y ~ x'



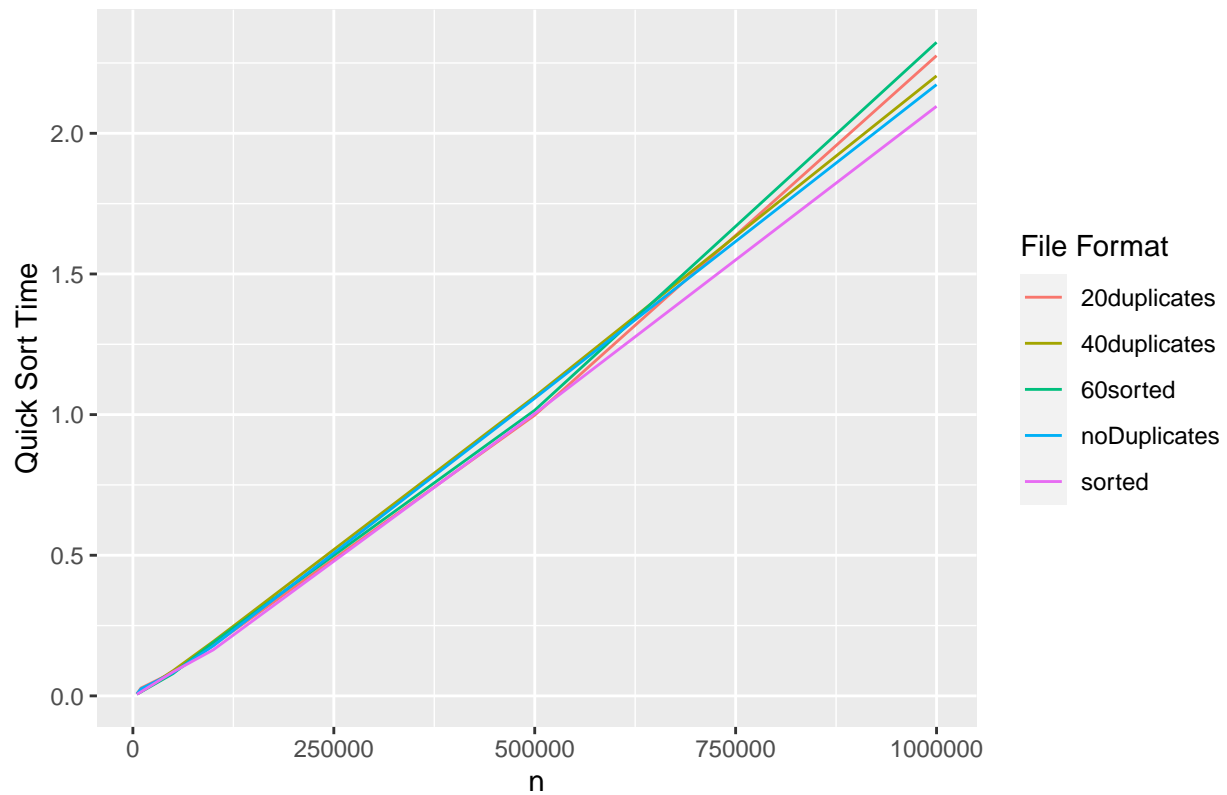
```
quickInts = subset(quickTimes, var_type == "int")
ggplot(quickInts, aes(x = size, y = quick_time, color = format)) +
  geom_line() +
  labs(title = "Quick Sort Time With Integer Data By Data Set Size and File Format", x = "n", y = "Quick")
  guides(color = guide_legend(title = "File Format"))
```

Quick Sort Time With Integer Data By Data Set Size and File Format



```
quickStrings = subset(quickTimes, var_type == "string")
ggplot(quickStrings, aes(x = size, y = quick_time, color = format)) +
  geom_line() +
  labs(title = "Quick Sort Time With String Data By Data Set Size and File Format", x = "n", y = "Quick")
guides(color = guide_legend(title = "File Format"))
```

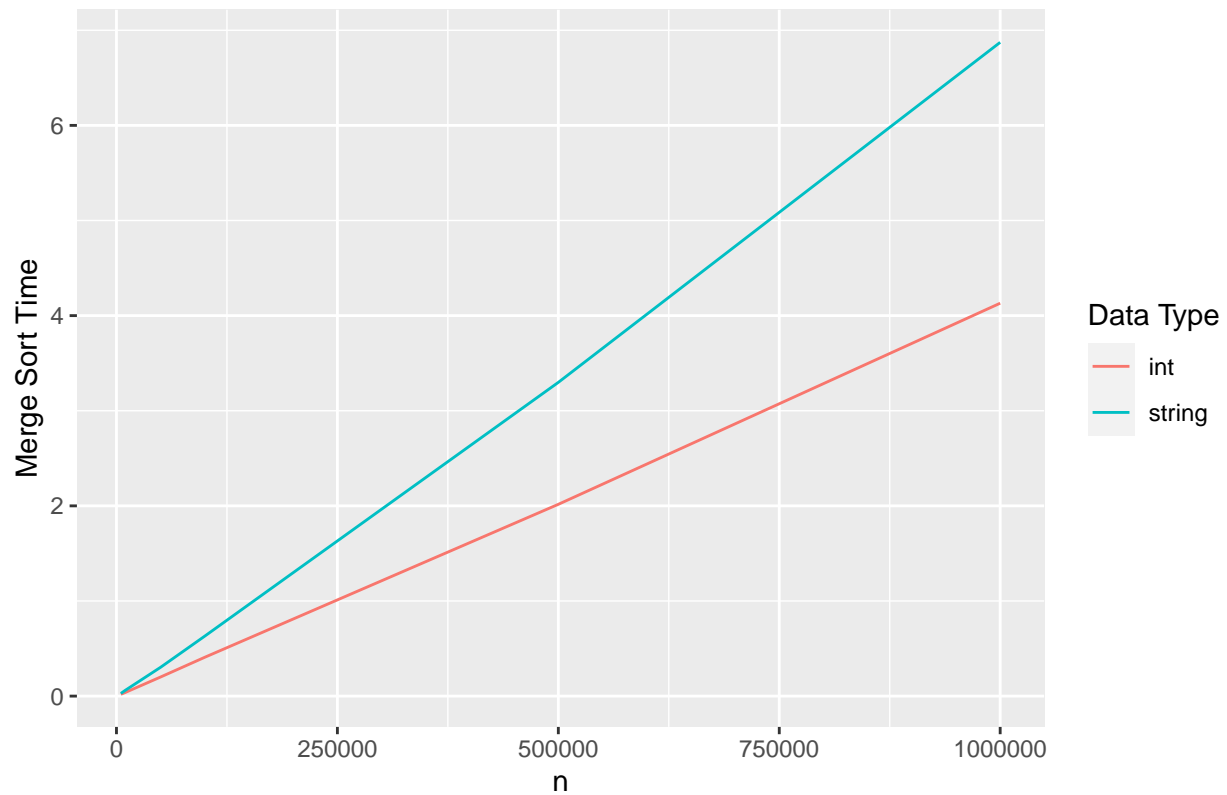
Quick Sort Time With String Data By Data Set Size and File Format



Merge Sort

```
mergeTimes = aggregate(merge_time ~ var_type + size + nlogn + format, data = data, FUN = mean)
mergeTimes2 = aggregate(merge_time ~ var_type + size + nlogn, data = data, FUN = mean)
ggplot(mergeTimes2, aes(x = size, y = merge_time, color = var_type)) +
  geom_line() +
  labs(title = "Mean Merge Sort Time By Data Set Size and Data Type", x = "n", y = "Merge Sort Time") +
  guides(color = guide_legend(title = "Data Type"))
```

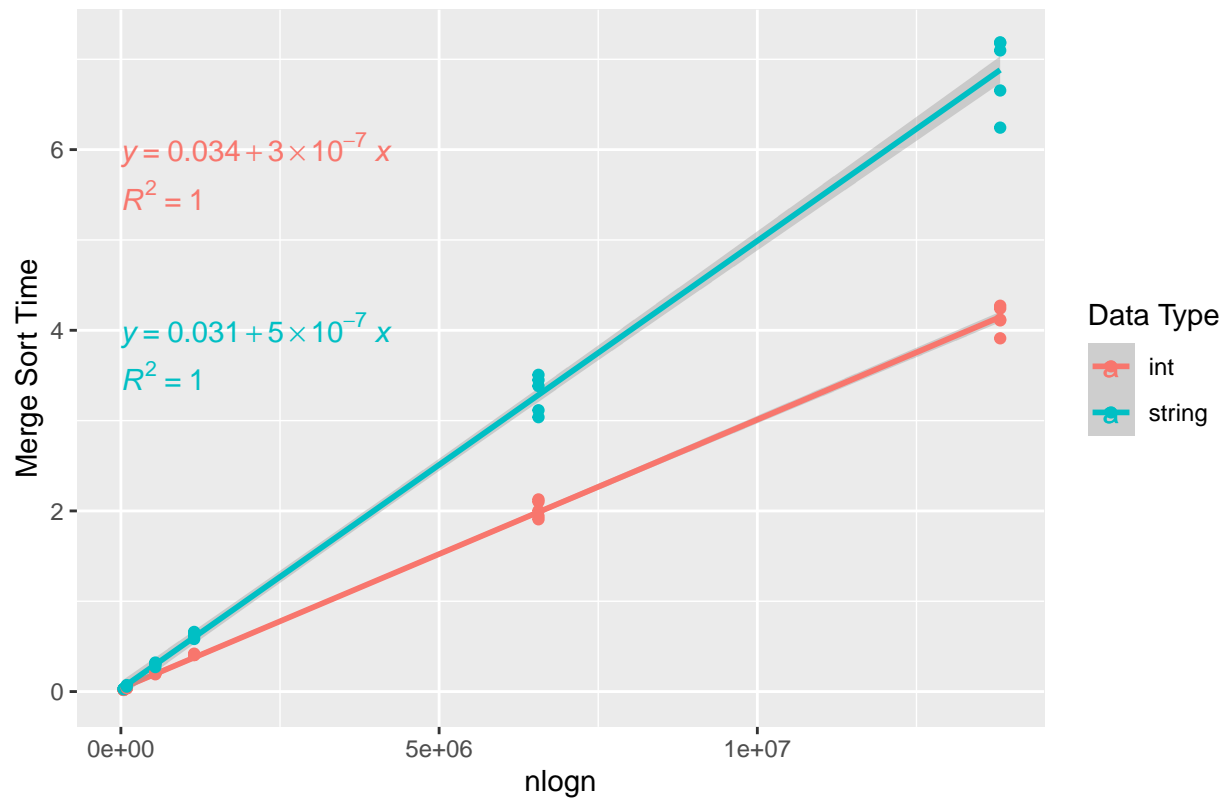
Mean Merge Sort Time By Data Set Size and Data Type



```
ggplot(mergeTimes, aes(x = nlogn, y = merge_time, color = var_type)) +
  labs(title = "Merge Sort Regression Models By Data Type", x = "nlogn", y = "Merge Sort Time") +
  geom_smooth(method="lm") +
  geom_point() +
  stat_regline_equation(label.x=0, label.y=c(6, 4)) +
  stat_cor(aes(label=..rr.label..), label.x=0, label.y=c(5.5, 3.5)) +
  guides(color = guide_legend(title = "Data Type"))
```

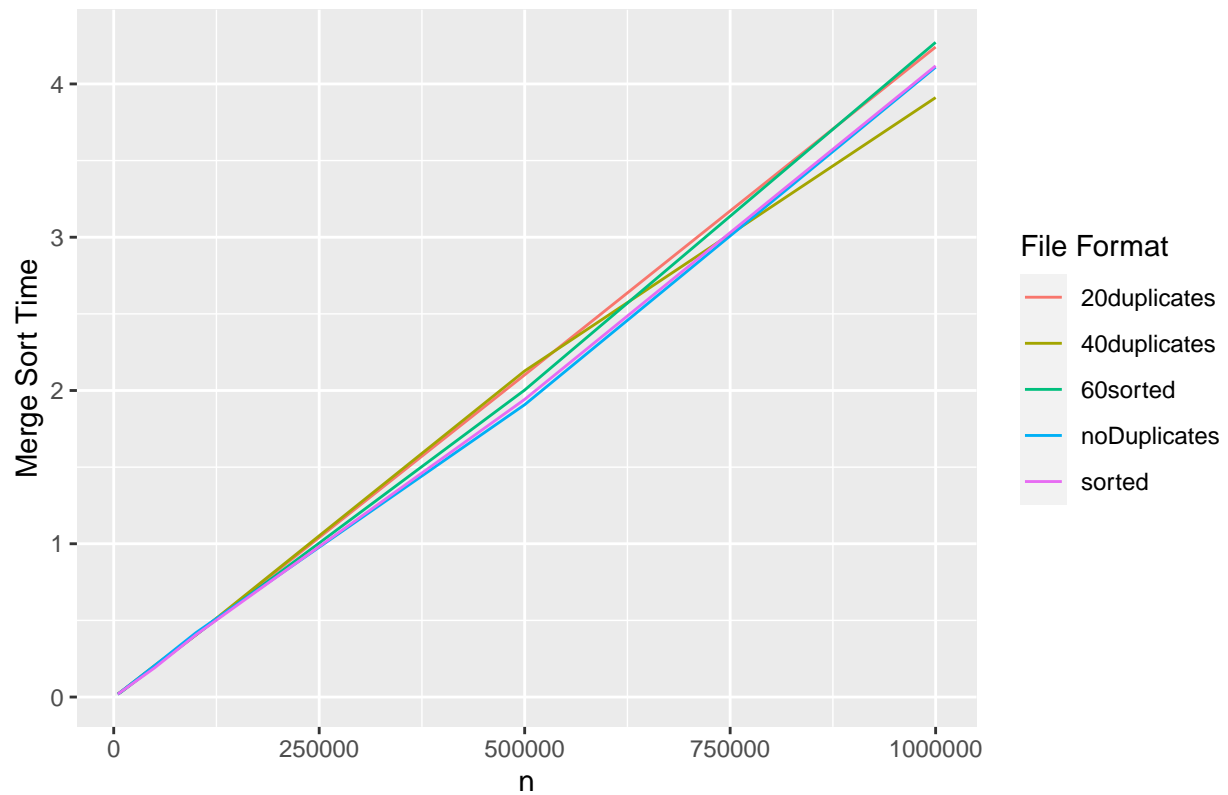
```
## 'geom_smooth()' using formula 'y ~ x'
```


Merge Sort Regression Models By Data Type



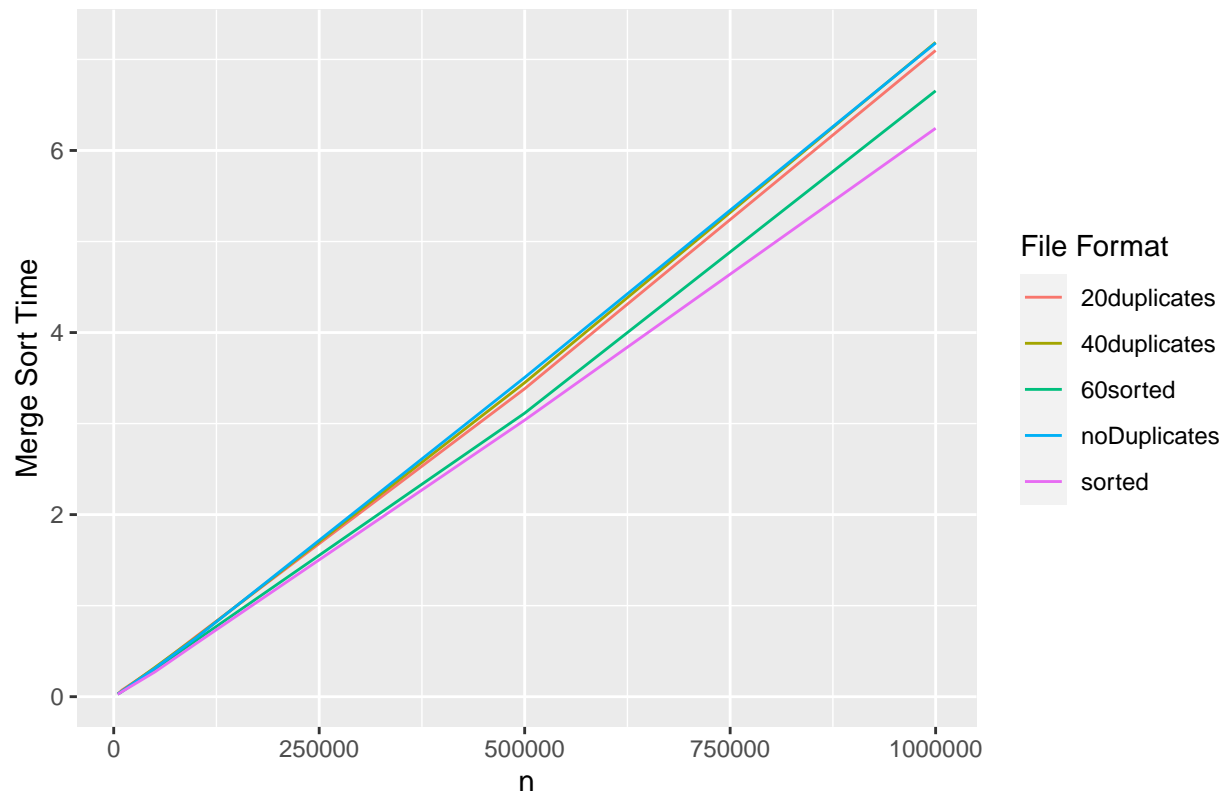
```
mergeInts = subset(mergeTimes, var_type == "int")
ggplot(mergeInts, aes(x = size, y = merge_time, color = format)) +
  geom_line() +
  labs(title = "Merge Sort Time With Integer Data By Data Set Size and File Format", x = "n", y = "Merge")
  guides(color = guide_legend(title = "File Format"))
```

Merge Sort Time With Integer Data By Data Set Size and File Format



```
mergeStrings = subset(mergeTimes, var_type == "string")
ggplot(mergeStrings, aes(x = size, y = merge_time, color = format)) +
  geom_line() +
  labs(title = "Merge Sort Time With String Data By Data Set Size and File Format", x = "n", y = "Merge")
guides(color = guide_legend(title = "File Format"))
```

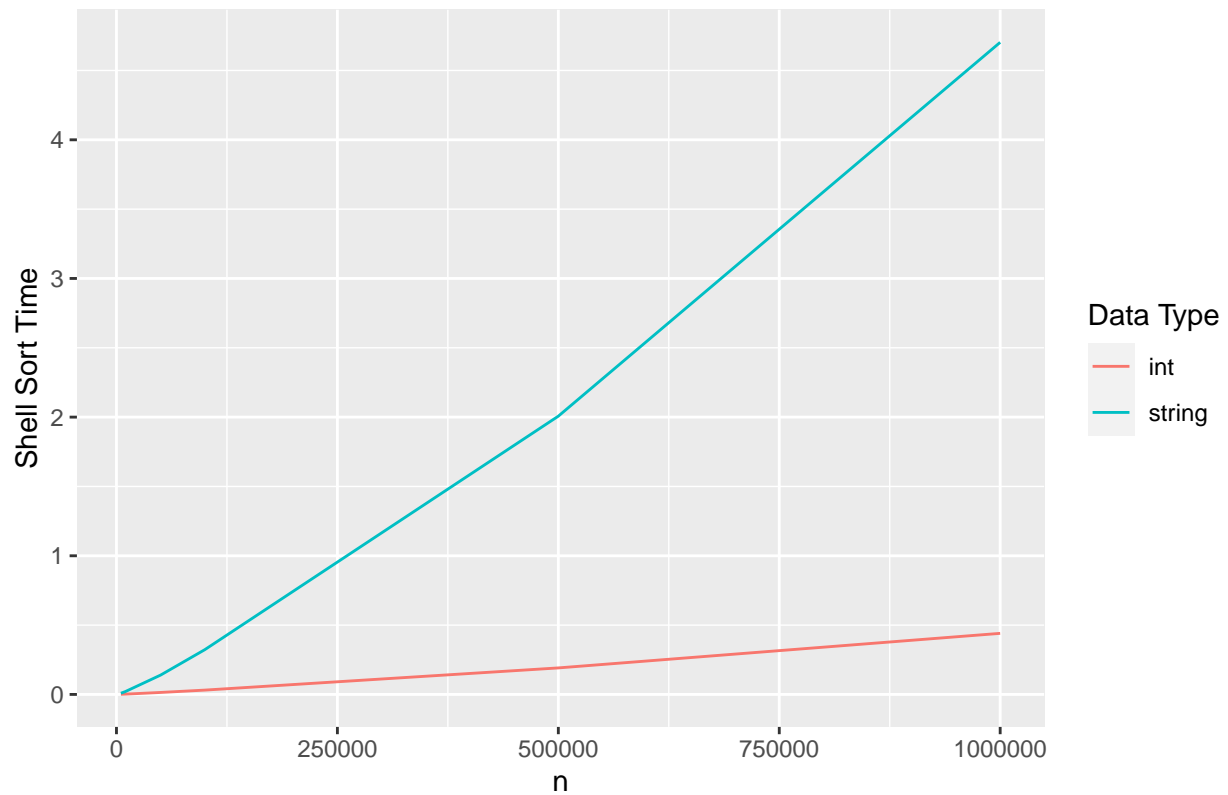
Merge Sort Time With String Data By Data Set Size and File Format



Shell Sort

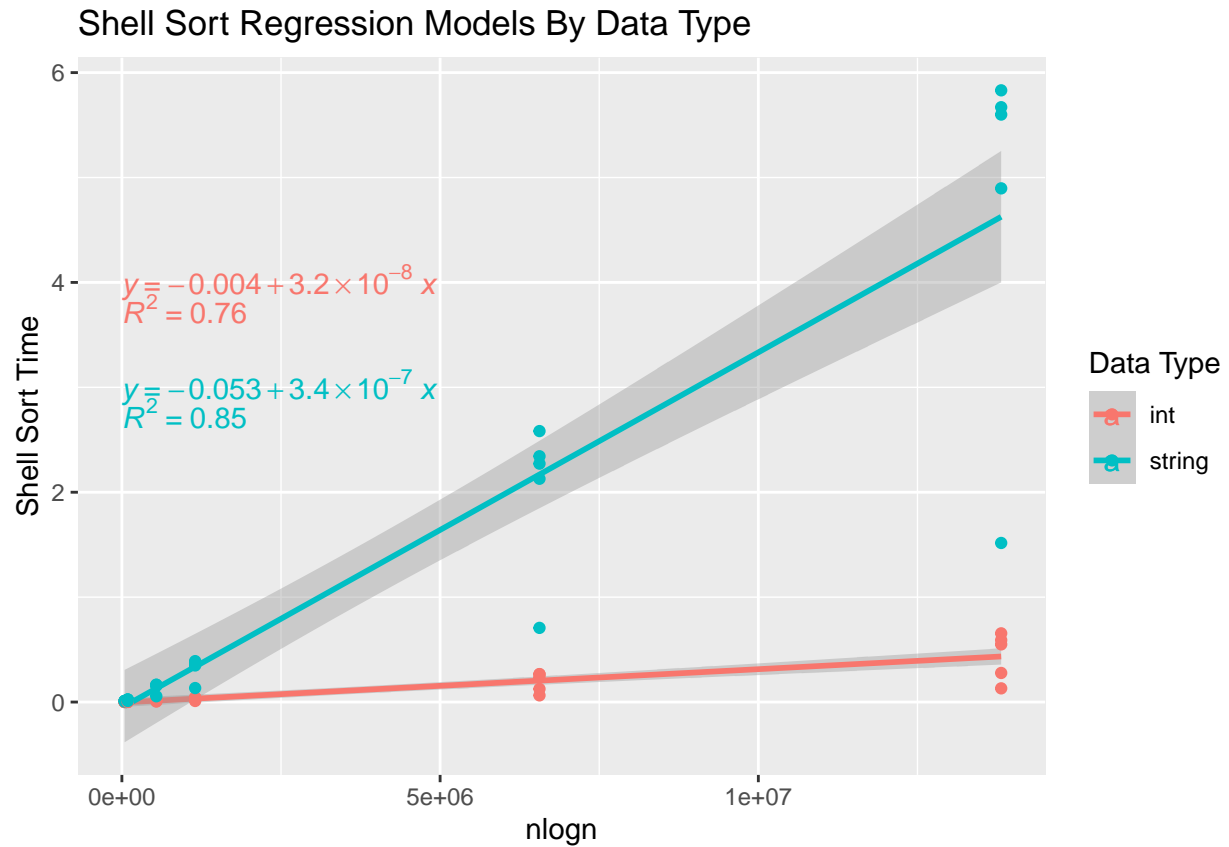
```
shellTimes = aggregate(shell_time ~ var_type + size + nlogn + format, data = data, FUN = mean)
shellTimes2 = aggregate(shell_time ~ var_type + size + nlogn, data = data, FUN = mean)
ggplot(shellTimes2, aes(x = size, y = shell_time, color = var_type)) +
  geom_line() +
  labs(title = "Mean Shell Sort Time By Data Set Size and Data Type", x = "n", y = "Shell Sort Time") +
  guides(color = guide_legend(title = "Data Type"))
```

Mean Shell Sort Time By Data Set Size and Data Type



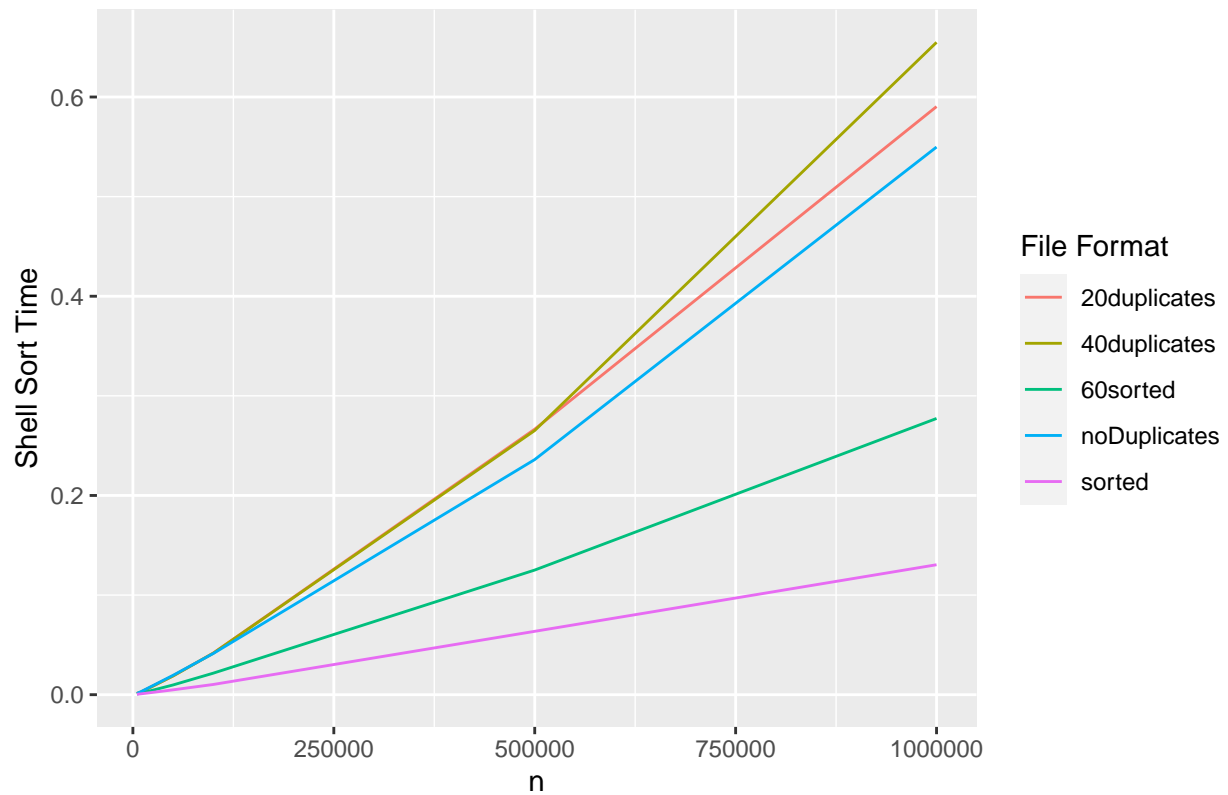
```
ggplot(shellTimes, aes(x = nlogn, y = shell_time, color = var_type)) +  
  labs(title = "Shell Sort Regression Models By Data Type", x = "nlogn", y = "Shell Sort Time") +  
  geom_smooth(method="lm") +  
  geom_point() +  
  stat_regline_equation(label.x=0, label.y=c(4, 3)) +  
  stat_cor(aes(label=..rr.label..), label.x=0, label.y=c(3.75, 2.75)) +  
  guides(color = guide_legend(title = "Data Type"))
```

```
## 'geom_smooth()' using formula 'y ~ x'
```

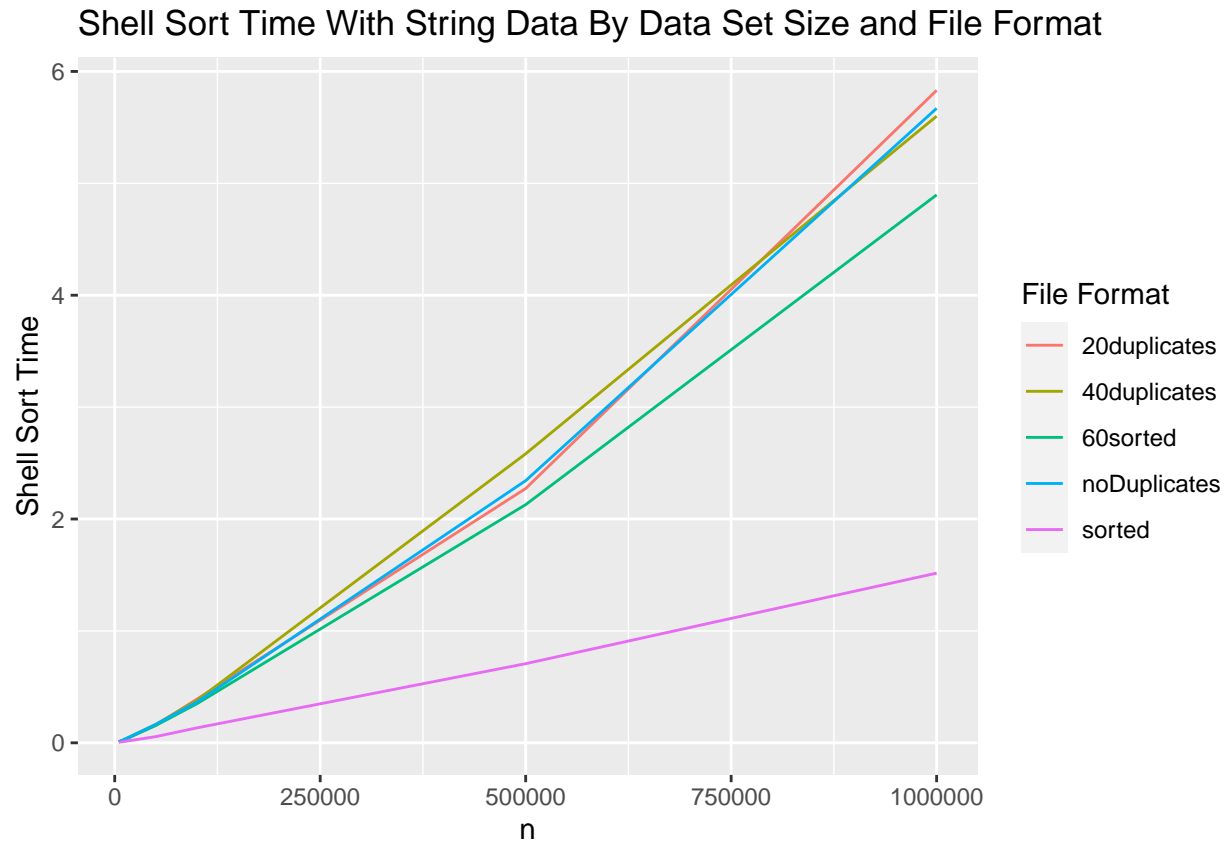


```
shellInts = subset(shellTimes, var_type == "int")
ggplot(shellInts, aes(x = size, y = shell_time, color = format)) +
  geom_line() +
  labs(title = "Shell Sort Time With Integer Data By Data Set Size and File Format", x = "n", y = "Shell")
guides(color = guide_legend(title = "File Format"))
```

Shell Sort Time With Integer Data By Data Set Size and File Format



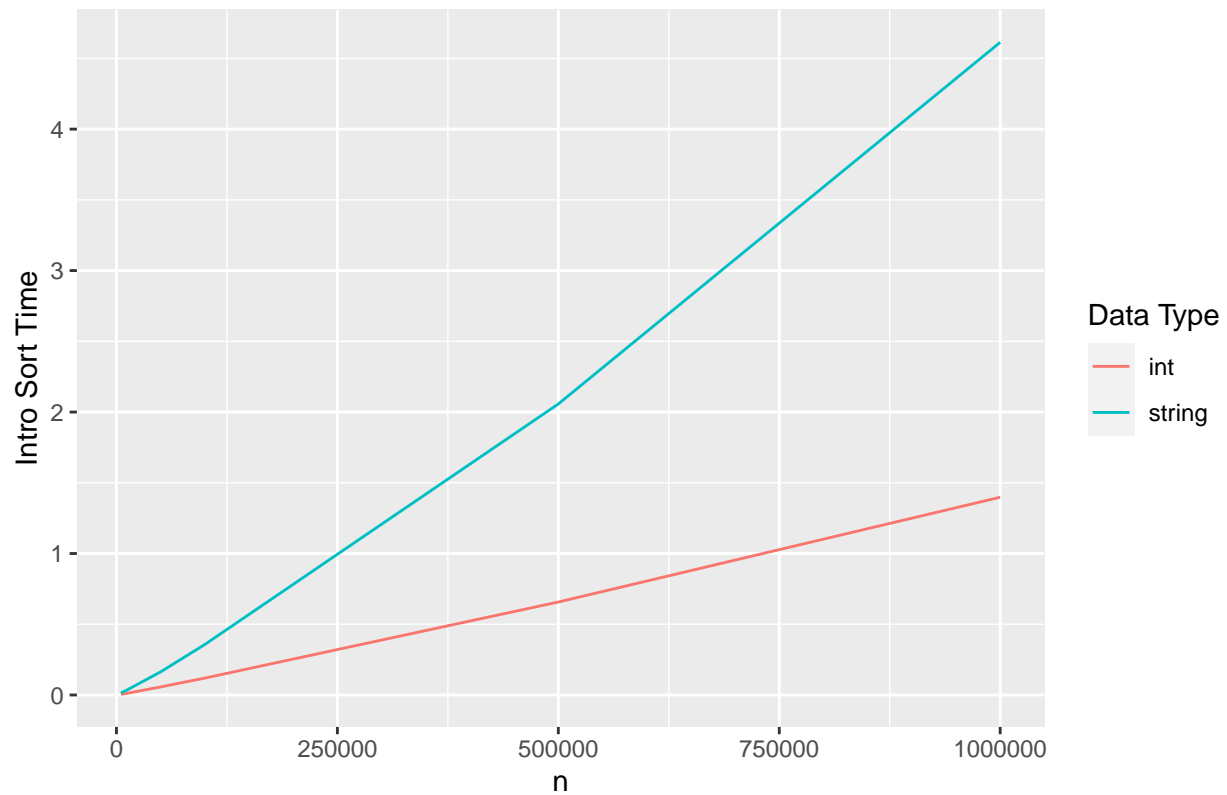
```
shellStrings = subset(shellTimes, var_type == "string")
ggplot(shellStrings, aes(x = size, y = shell_time, color = format)) +
  geom_line() +
  labs(title = "Shell Sort Time With String Data By Data Set Size and File Format", x = "n", y = "Shell")
guides(color = guide_legend(title = "File Format"))
```



Intro Sort

```
introTimes = aggregate(intro_time ~ var_type + size + nlogn + format, data = data, FUN = mean)
introTimes2 = aggregate(intro_time ~ var_type + size + nlogn, data = data, FUN = mean)
ggplot(introTimes2, aes(x = size, y = intro_time, color = var_type)) +
  geom_line() +
  labs(title = "Mean Intro Sort Time By Data Set Size and Data Type", x = "n", y = "Intro Sort Time") +
  guides(color = guide_legend(title = "Data Type"))
```

Mean Intro Sort Time By Data Set Size and Data Type



```
ggplot(introTimes, aes(x = nlogn, y = intro_time, color = var_type)) +  
  labs(title = "Intro Sort Regression Models By Data Type", x = "nlogn", y = "Intro Sort Time") +  
  geom_smooth(method="lm") +  
  geom_point() +  
  stat_regline_equation(label.x=0, label.y=c(4, 3)) +  
  stat_cor(aes(label=..rr.label..), label.x=0, label.y=c(3.75, 2.75)) +  
  guides(color = guide_legend(title = "Data Type"))
```

```
## 'geom_smooth()' using formula 'y ~ x'
```




```
introInts = subset(introTimes, var_type == "int")
ggplot(introInts, aes(x = size, y = intro_time, color = format)) +
  geom_line() +
  labs(title = "Intro Sort Time With Integer Data By Data Set Size and File Format", x = "n", y = "Intro")
  guides(color = guide_legend(title = "File Format"))
```



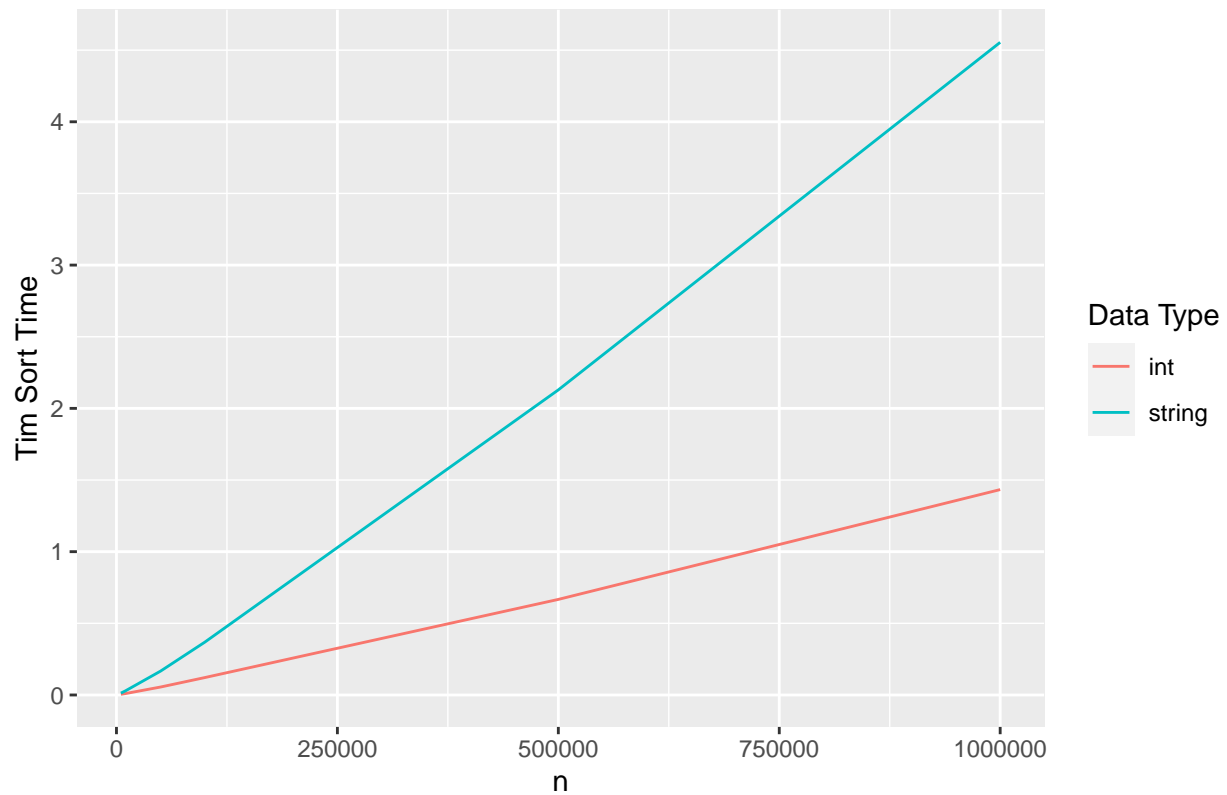
```
introStrings = subset(introTimes, var_type == "string")
ggplot(introStrings, aes(x = size, y = intro_time, color = format)) +
  geom_line() +
  labs(title = "Intro Sort Time With String Data By Data Set Size and File Format", x = "n", y = "Intro")
guides(color = guide_legend(title = "File Format"))
```



Tim Sort

```
timTimes = aggregate(tim_time ~ var_type + size + nlogn + format, data = data, FUN = mean)
timTimes2 = aggregate(tim_time ~ var_type + size + nlogn, data = data, FUN = mean)
ggplot(timTimes2, aes(x = size, y = tim_time, color = var_type)) +
  geom_line() +
  labs(title = "Mean Tim Sort Time By Data Set Size and Data Type", x = "n", y = "Tim Sort Time") +
  guides(color = guide_legend(title = "Data Type"))
```

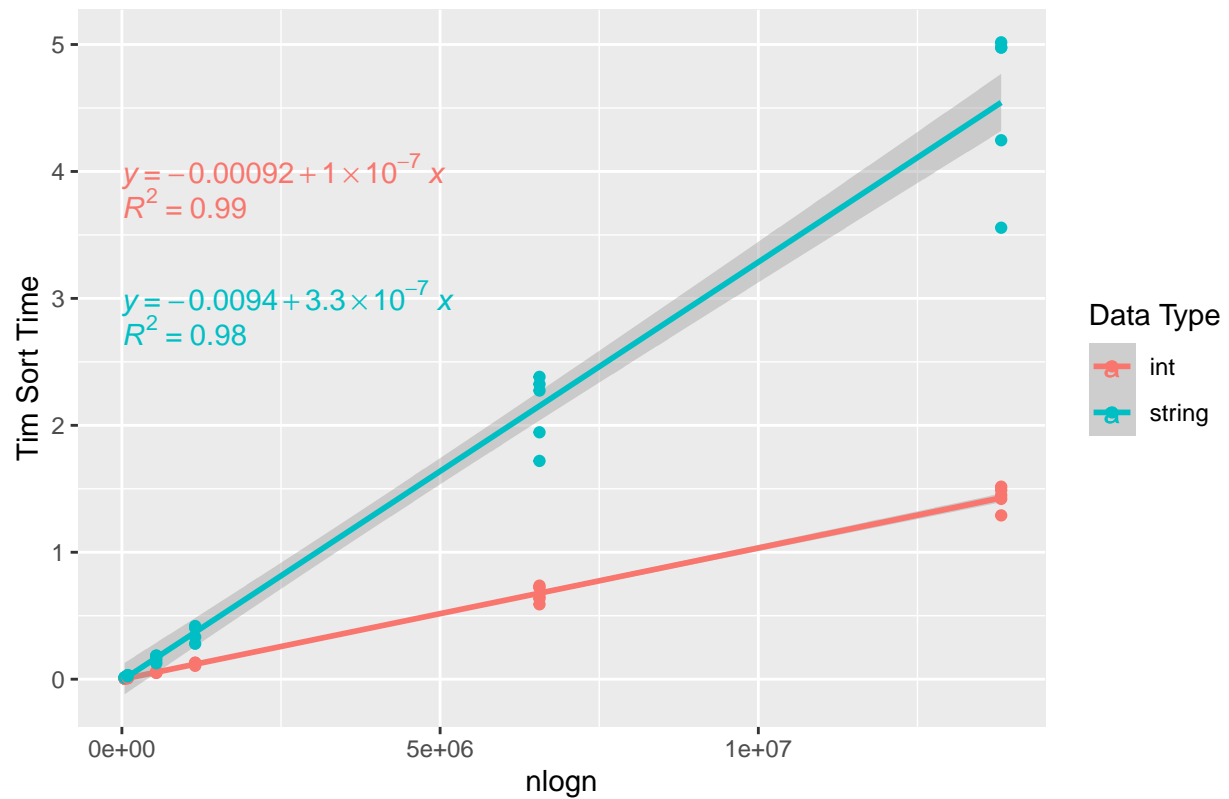
Mean Tim Sort Time By Data Set Size and Data Type



```
ggplot(timTimes, aes(x = nlogn, y = tim_time, color = var_type)) +  
  labs(title = "Tim Sort Regression Models By Data Type", x = "nlogn", y = "Tim Sort Time") +  
  geom_smooth(method="lm") +  
  geom_point() +  
  stat_regline_equation(label.x=0, label.y=c(4, 3)) +  
  stat_cor(aes(label=..rr.label..), label.x=0, label.y=c(3.75, 2.75)) +  
  guides(color = guide_legend(title = "Data Type"))
```

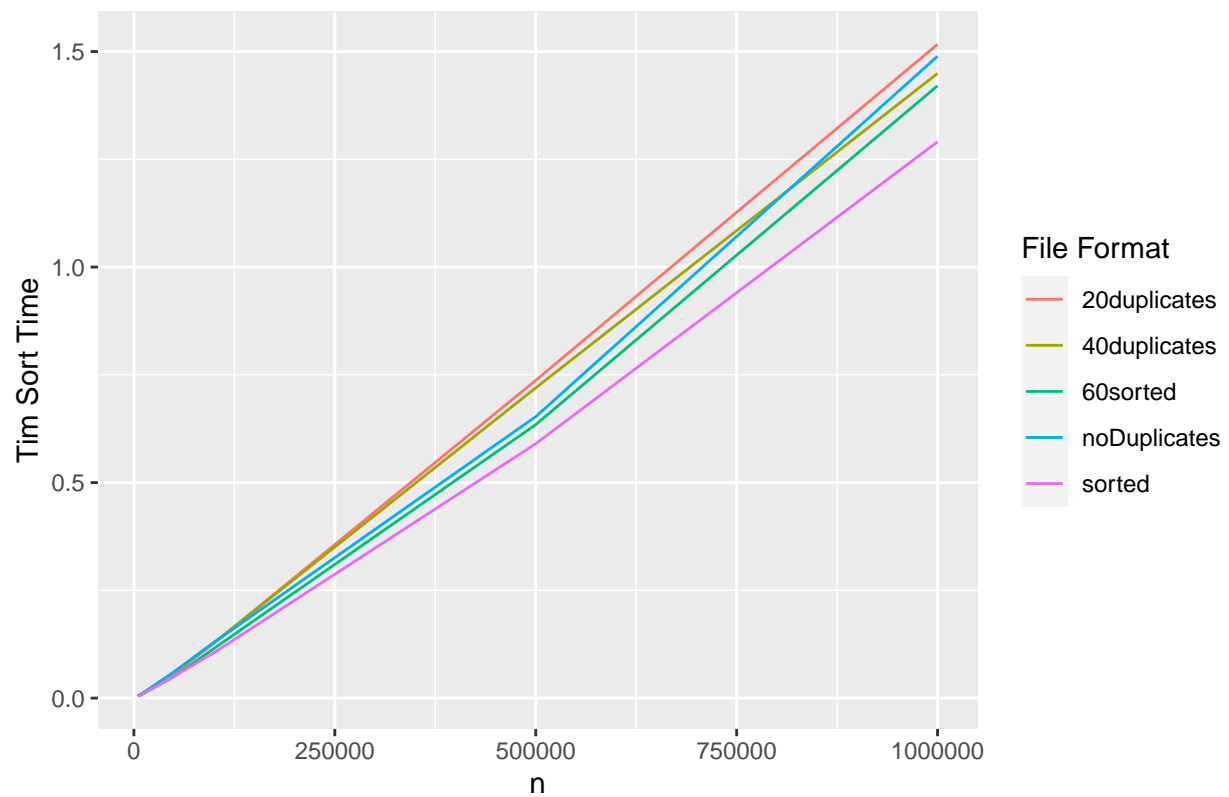
```
## 'geom_smooth()' using formula 'y ~ x'
```

Tim Sort Regression Models By Data Type



```
timInts = subset(timTimes, var_type == "int")
ggplot(timInts, aes(x = size, y = tim_time, color = format)) +
  geom_line() +
  labs(title = "Tim Sort Time With Integer Data By Data Set Size and File Format", x = "n", y = "Tim So")
  guides(color = guide_legend(title = "File Format"))
```

Tim Sort Time With Integer Data By Data Set Size and File Format



```
timStrings = subset(timTimes, var_type == "string")
ggplot(timStrings, aes(x = size, y = tim_time, color = format)) +
  geom_line() +
  labs(title = "Tim Sort Time With String Data By Data Set Size and File Format", x = "n", y = "Tim Sort Time") +
  guides(color = guide_legend(title = "File Format"))
```

Tim Sort Time With String Data By Data Set Size and File Format

