

Program 2 Graph Analysis

Ryan Schaefer and Wes Anderson

Create Dataset

```
library(ggplot2)
library(ggpubr)
data = read.csv("RyanDataRun2.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	0.25182200	3.3929000
## 2	int	1000000	40duplicates	N/A	0.51712400	7.0445000
## 3	int	100000	40duplicates	N/A	0.04598970	0.6683090
## 4	int	10000	40duplicates	N/A	0.00404639	0.0619693
## 5	int	50000	sorted	N/A	0.01525070	0.3114110
## 6	int	50000	20duplicates	N/A	0.02176670	0.3224080
## 7	int	5000	noDuplicates	N/A	0.00188733	0.0298806
## 8	int	500000	sorted	N/A	0.18726900	3.2843200
## 9	int	500000	60sorted	N/A	0.22327600	3.2930900
## 10	int	10000	60sorted	N/A	0.00348286	0.0610733
## 11	int	1000000	noDuplicates	N/A	0.51682800	6.9283400
## 12	int	1000000	20duplicates	N/A	0.52818900	6.9518500
## 13	int	50000	noDuplicates	N/A	0.02209720	0.3219980
## 14	int	5000	60sorted	N/A	0.00153918	0.0297214
## 15	int	5000	sorted	N/A	0.00140858	0.0290343
## 16	int	100000	20duplicates	N/A	0.04545140	0.6627060
## 17	int	50000	60sorted	N/A	0.01778230	0.3157490
## 18	int	10000	noDuplicates	N/A	0.00397305	0.0616479
## 19	int	500000	20duplicates	N/A	0.24854200	3.3887300
## 20	int	500000	40duplicates	N/A	0.25068800	3.3957400
## 21	int	1000000	sorted	N/A	0.35124400	6.6323600
## 22	int	5000	20duplicates	N/A	0.00192626	0.0299672
## 23	int	100000	noDuplicates	N/A	0.04593360	0.6687710
## 24	int	50000	40duplicates	N/A	0.02204200	0.3299100
## 25	int	10000	20duplicates	N/A	0.00411101	0.0630750
## 26	int	100000	sorted	N/A	0.03047480	0.6483230
## 27	int	100000	60sorted	N/A	0.03967040	0.6594540
## 28	int	10000	sorted	N/A	0.00264716	0.0612166
## 29	int	5000	40duplicates	N/A	0.00187671	0.0307168
## 30	int	1000000	60sorted	N/A	0.43119800	6.7950700
## 31	string	50000	sorted	N/A	0.12517500	0.5155900
## 32	string	500000	20duplicates	N/A	1.73995000	6.3040100
## 33	string	50000	20duplicates	N/A	0.14014000	0.5826260

## 34	string	10000	40duplicates	N/A	0.02370550	0.1058390
## 35	string	10000	60sorted	N/A	0.02522590	0.0999545
## 36	string	100000	sorted	N/A	0.28516300	1.0563800
## 37	string	5000	40duplicates	N/A	0.01143800	0.0506749
## 38	string	500000	60sorted	N/A	1.75721000	5.9481400
## 39	string	50000	noDuplicates	N/A	0.14068200	0.6115850
## 40	string	500000	40duplicates	N/A	1.80567000	6.4913000
## 41	string	5000	20duplicates	N/A	0.01180990	0.0535724
## 42	string	100000	noDuplicates	N/A	0.30530500	1.2381700
## 43	string	5000	noDuplicates	N/A	0.01069510	0.0526140
## 44	string	100000	60sorted	N/A	0.30960300	1.1733500
## 45	string	1000000	20duplicates	N/A	3.81779000	14.1716000
## 46	string	10000	noDuplicates	N/A	0.02371090	0.1078230
## 47	string	1000000	noDuplicates	N/A	3.72267000	13.7663000
## 48	string	1000000	sorted	N/A	3.53418000	11.7874000
## 49	string	500000	noDuplicates	N/A	1.79320000	6.4953400
## 50	string	100000	40duplicates	N/A	0.32554100	1.3067900
## 51	string	5000	60sorted	N/A	0.01090460	0.0555941
## 52	string	1000000	60sorted	N/A	3.96313000	12.3459000
## 53	string	5000	sorted	N/A	0.01099300	0.0504623
## 54	string	100000	20duplicates	N/A	0.30769300	1.2787500
## 55	string	10000	20duplicates	N/A	0.02464090	0.1065780
## 56	string	10000	sorted	N/A	0.02326950	0.1155710
## 57	string	500000	sorted	N/A	1.64137000	5.8828400
## 58	string	50000	40duplicates	N/A	0.15211700	0.5875090
## 59	string	50000	60sorted	N/A	0.14217500	0.6274830
## 60	string	1000000	40duplicates	N/A	3.63300000	13.5628000
##	shell_time	intro_time	tim_time	n2	nlogn	
## 1	4.65442e-01	1.22008000	1.29866000	2.5e+11	6561181.69	
## 2	1.02661e+00	2.60419000	2.74607000	1.0e+12	13815510.56	
## 3	7.28055e-02	0.23186500	0.23424200	1.0e+10	1151292.55	
## 4	4.74999e-03	0.01732980	0.01904800	1.0e+08	92103.40	
## 5	9.04208e-03	0.10095600	0.09160940	2.5e+09	540988.91	
## 6	3.21098e-02	0.10408700	0.10805400	2.5e+09	540988.91	
## 7	2.17432e-03	0.00784260	0.00906813	2.5e+07	42585.97	
## 8	1.12102e-01	1.14242000	1.10672000	2.5e+11	6561181.69	
## 9	2.27612e-01	1.14562000	1.15008000	2.5e+11	6561181.69	
## 10	2.65781e-03	0.01743650	0.01728360	1.0e+08	92103.40	
## 11	1.02587e+00	2.52421000	2.69509000	1.0e+12	13815510.56	
## 12	1.04653e+00	2.58902000	2.69777000	1.0e+12	13815510.56	
## 13	3.30349e-02	0.10111400	0.10790600	2.5e+09	540988.91	
## 14	1.21408e-03	0.00799931	0.01115470	2.5e+07	42585.97	
## 15	7.07495e-04	0.00796457	0.00760660	2.5e+07	42585.97	
## 16	7.20035e-02	0.21883600	0.23276800	1.0e+10	1151292.55	
## 17	1.74423e-02	0.10206900	0.09789040	2.5e+09	540988.91	
## 18	4.91543e-03	0.01686420	0.01928410	1.0e+08	92103.40	
## 19	4.62115e-01	1.20079000	1.27915000	2.5e+11	6561181.69	
## 20	4.63212e-01	1.19731000	1.27181000	2.5e+11	6561181.69	
## 21	2.38091e-01	2.41628000	2.29527000	1.0e+12	13815510.56	
## 22	2.11924e-03	0.00802482	0.00892325	2.5e+07	42585.97	
## 23	7.30912e-02	0.21668100	0.23826700	1.0e+10	1151292.55	
## 24	3.29287e-02	0.10447800	0.11097300	2.5e+09	540988.91	
## 25	4.88992e-03	0.01886800	0.01950070	1.0e+08	92103.40	
## 26	2.00971e-02	0.19966900	0.19869900	1.0e+10	1151292.55	

```
## 27 3.85674e-02 0.21785300 0.21290100 1.0e+10 1151292.55
## 28 1.55198e-03 0.01823090 0.01636600 1.0e+08 92103.40
## 29 2.16847e-03 0.00809562 0.00905273 2.5e+07 42585.97
## 30 5.01503e-01 2.49849000 2.44505000 1.0e+12 13815510.56
## 31 1.15221e-01 0.25012400 0.25133900 2.5e+09 540988.91
## 32 4.35572e+00 3.55594000 4.45394000 2.5e+11 6561181.69
## 33 3.00520e-01 0.30899400 0.37720700 2.5e+09 540988.91
## 34 4.26276e-02 0.05002950 0.06519150 1.0e+08 92103.40
## 35 4.19649e-02 0.04962450 0.05173980 1.0e+08 92103.40
## 36 2.47682e-01 0.59598400 0.55122800 1.0e+10 1151292.55
## 37 1.82914e-02 0.02688160 0.03000000 2.5e+07 42585.97
## 38 4.01005e+00 3.55840000 3.69578000 2.5e+11 6561181.69
## 39 3.03076e-01 0.29746100 0.37430500 2.5e+09 540988.91
## 40 4.82374e+00 3.71235000 4.49199000 2.5e+11 6561181.69
## 41 1.83925e-02 0.02787970 0.03081080 2.5e+07 42585.97
## 42 6.82243e-01 0.66146400 0.79729500 1.0e+10 1151292.55
## 43 1.85567e-02 0.02154240 0.03378680 2.5e+07 42585.97
## 44 6.57772e-01 0.65389100 0.65396800 1.0e+10 1151292.55
## 45 1.08192e+01 7.99185000 9.48475000 1.0e+12 13815510.56
## 46 4.40620e-02 0.06157530 0.06752750 1.0e+08 92103.40
## 47 1.06312e+01 8.08455000 9.39056000 1.0e+12 13815510.56
## 48 2.96448e+00 7.64843000 6.57742000 1.0e+12 13815510.56
## 49 4.43171e+00 4.03731000 4.60732000 2.5e+11 6561181.69
## 50 7.04154e-01 0.72503900 0.81573000 1.0e+10 1151292.55
## 51 1.68940e-02 0.02279240 0.02784000 2.5e+07 42585.97
## 52 9.02456e+00 8.15409000 7.71092000 1.0e+12 13815510.56
## 53 9.05521e-03 0.02099510 0.01964520 2.5e+07 42585.97
## 54 7.06555e-01 0.67575100 0.84747100 1.0e+10 1151292.55
## 55 4.31165e-02 0.05554790 0.07073060 1.0e+08 92103.40
## 56 2.01136e-02 0.05259610 0.05616930 1.0e+08 92103.40
## 57 1.41143e+00 3.68321000 3.09502000 2.5e+11 6561181.69
## 58 3.03167e-01 0.33351800 0.37739100 2.5e+09 540988.91
## 59 2.82605e-01 0.34391600 0.30525500 2.5e+09 540988.91
## 60 1.06043e+01 8.52516000 9.38546000 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"))
```

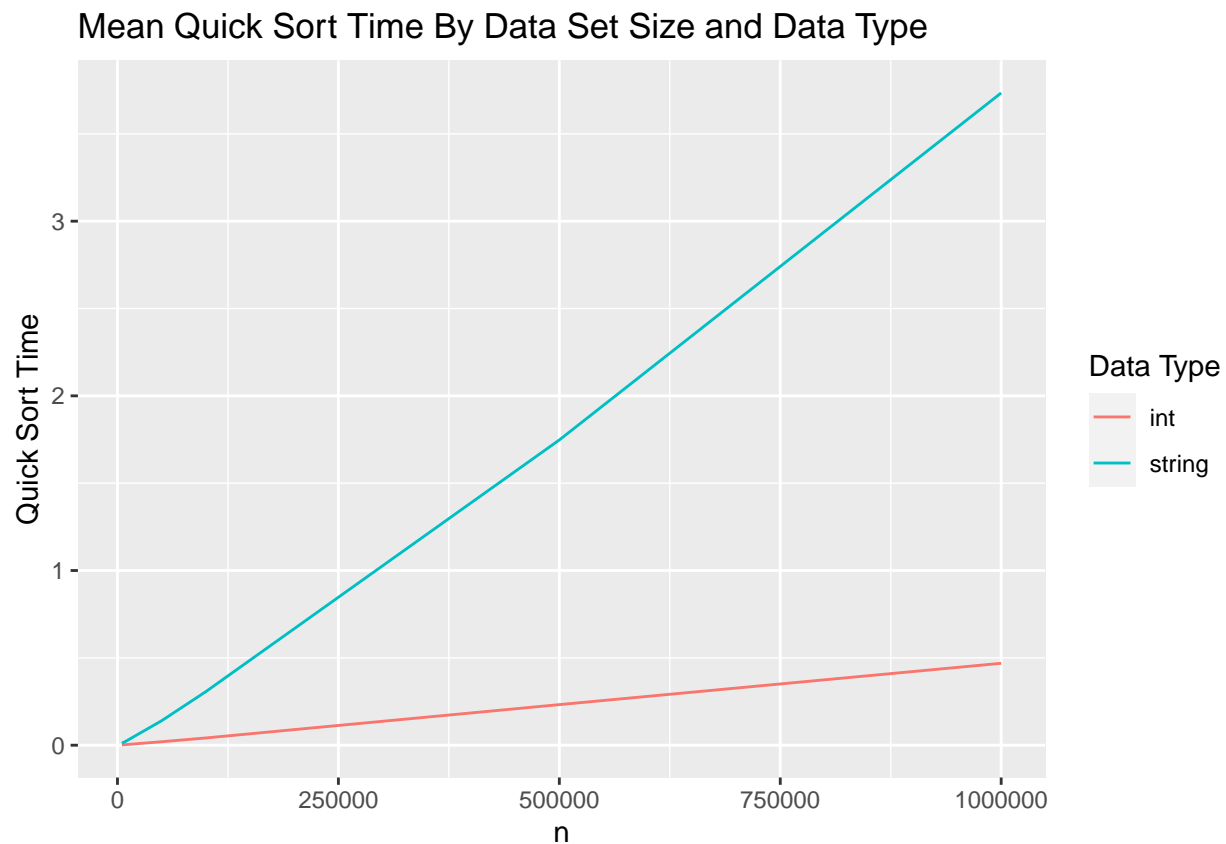
```
# ggplot(insertionTimes, aes(x = size, y = insertion_time, color = var_type)) +
#   labs(title = "Insertion Sort Regression Models By Data Type", x = "n^2", y = "Insertion Sort Time")
#   geom_smooth(method="lm") +
#   geom_point() +
#   stat_regline_equation(label.x=0, label.y=c(9000, 6000)) +
#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



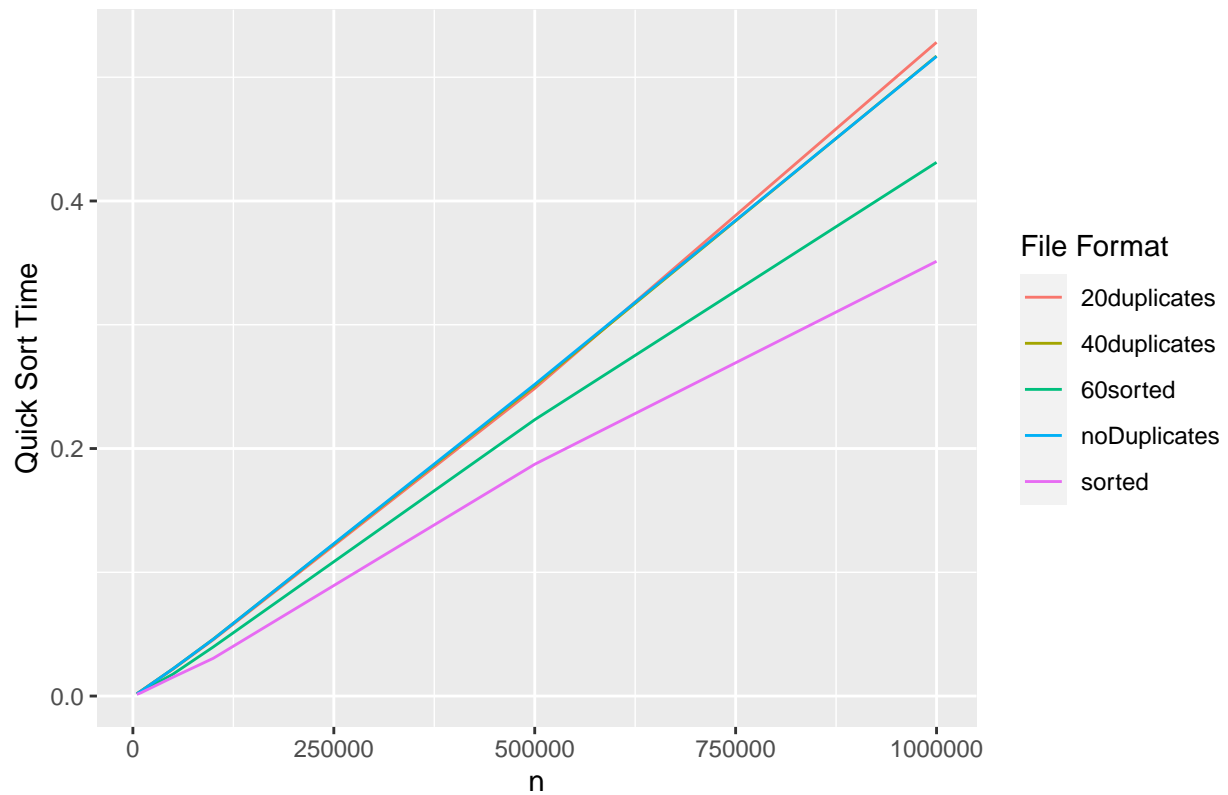
```
ggplot(quickTimes, aes(x = nlogn, y = quick_time, color = var_type)) +
  labs(title = "Quick Sort Regression Models By Data Type", x = "nlogn", y = "Insertion Sort Time") +
  geom_smooth(method="lm") +
  geom_point() +
  stat_regline_equation(label.x=0, label.y=c(1.5, 1)) +
  stat_cor(aes(label=..rr.label..),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

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



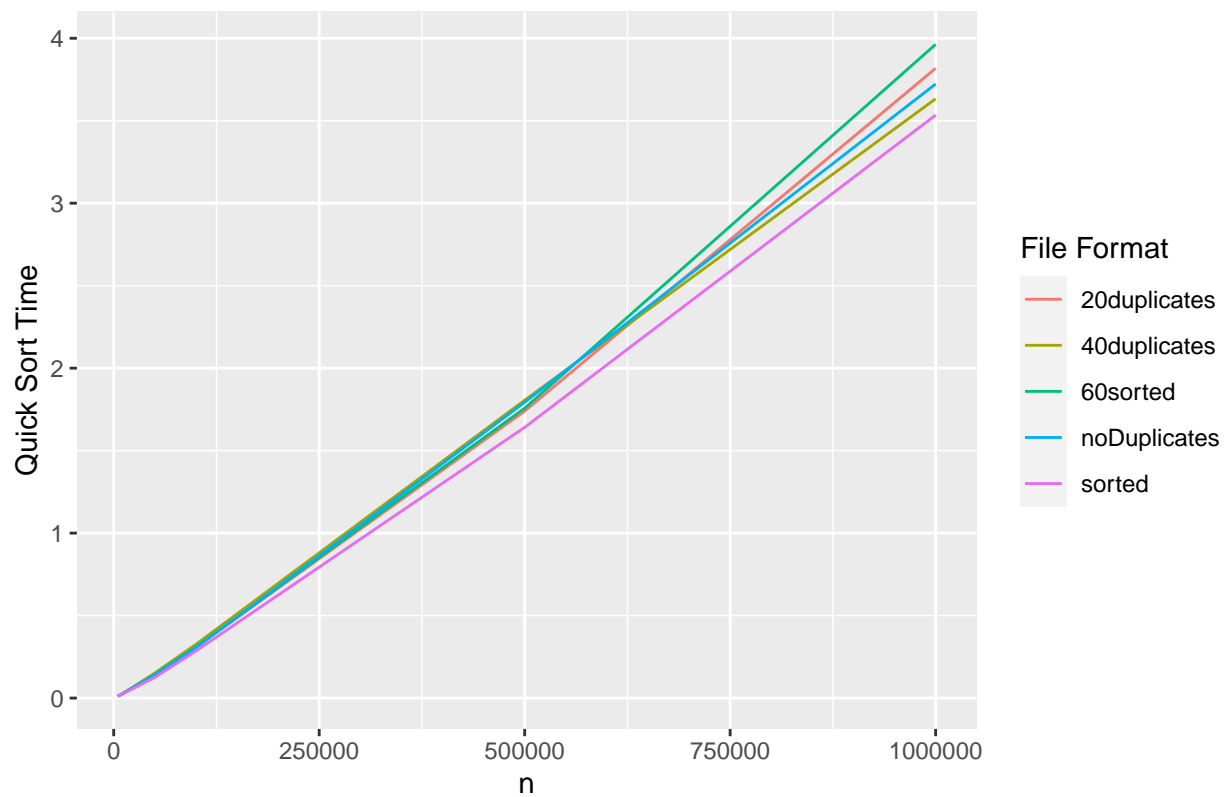
```
quickInts = subset(quickTimes, var_type == "int")
ggplot(quickInts, aes(x = size, y = quick_time, color = format)) +
  geom_line() +
  labs(title = "Quick Sort Time With Integer Data By Data Set Size and File Format", x = "n", y = "Quick")
  guides(color = guide_legend(title = "File Format"))
```

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"))
```

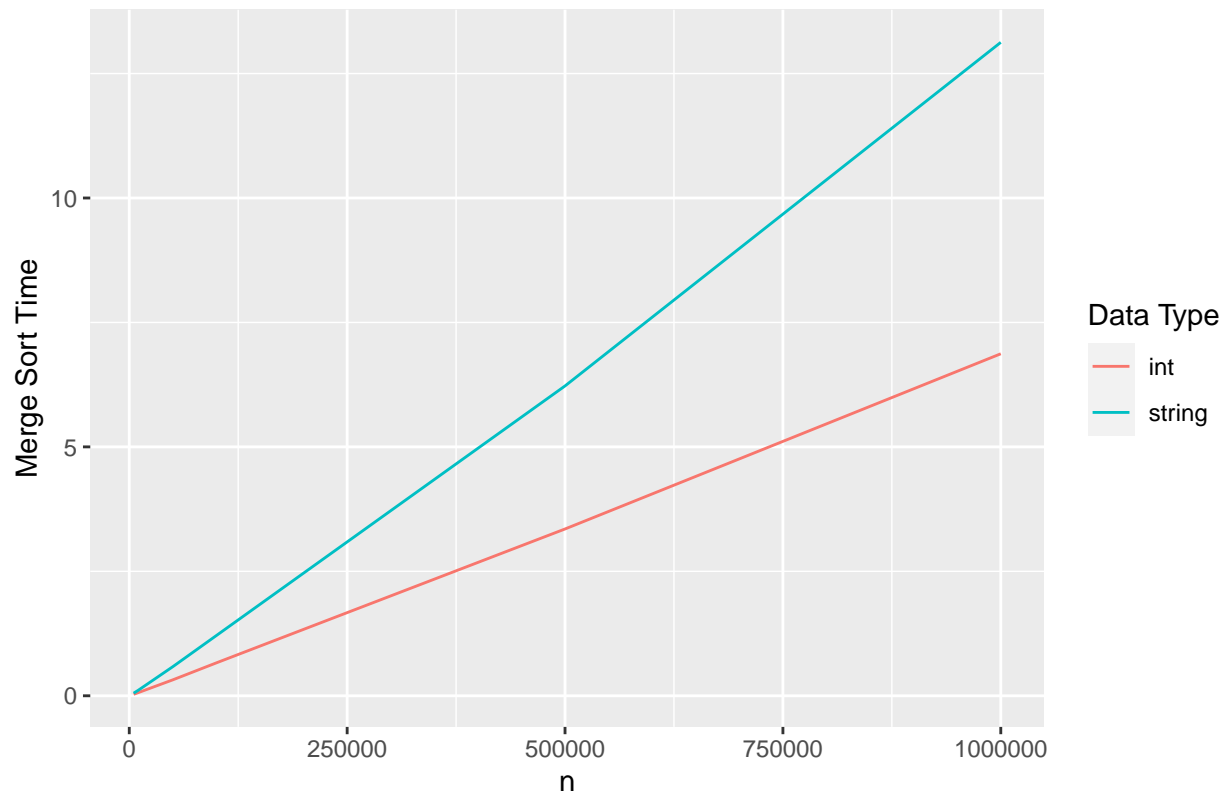
Quick Sort Time With String Data By Data Set Size and File Format



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"))
```

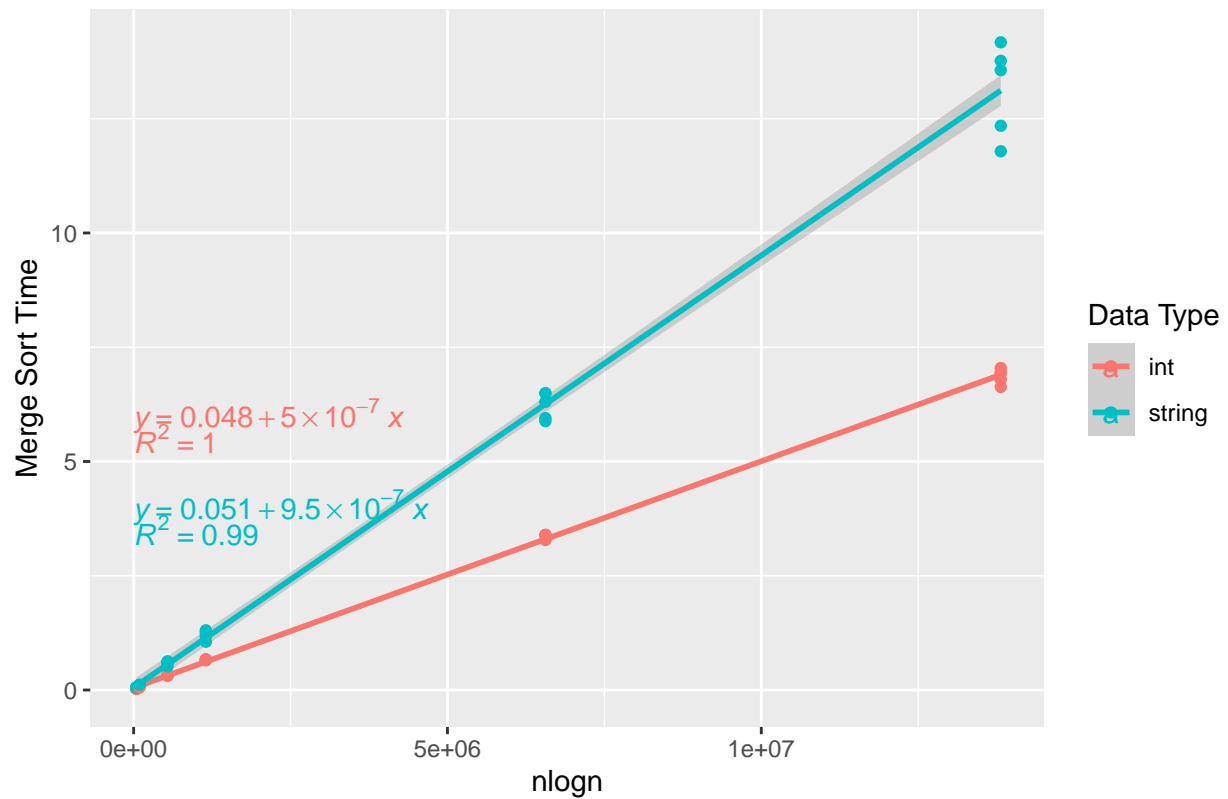
Mean Merge Sort Time By Data Set Size and Data Type



```
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"))
```

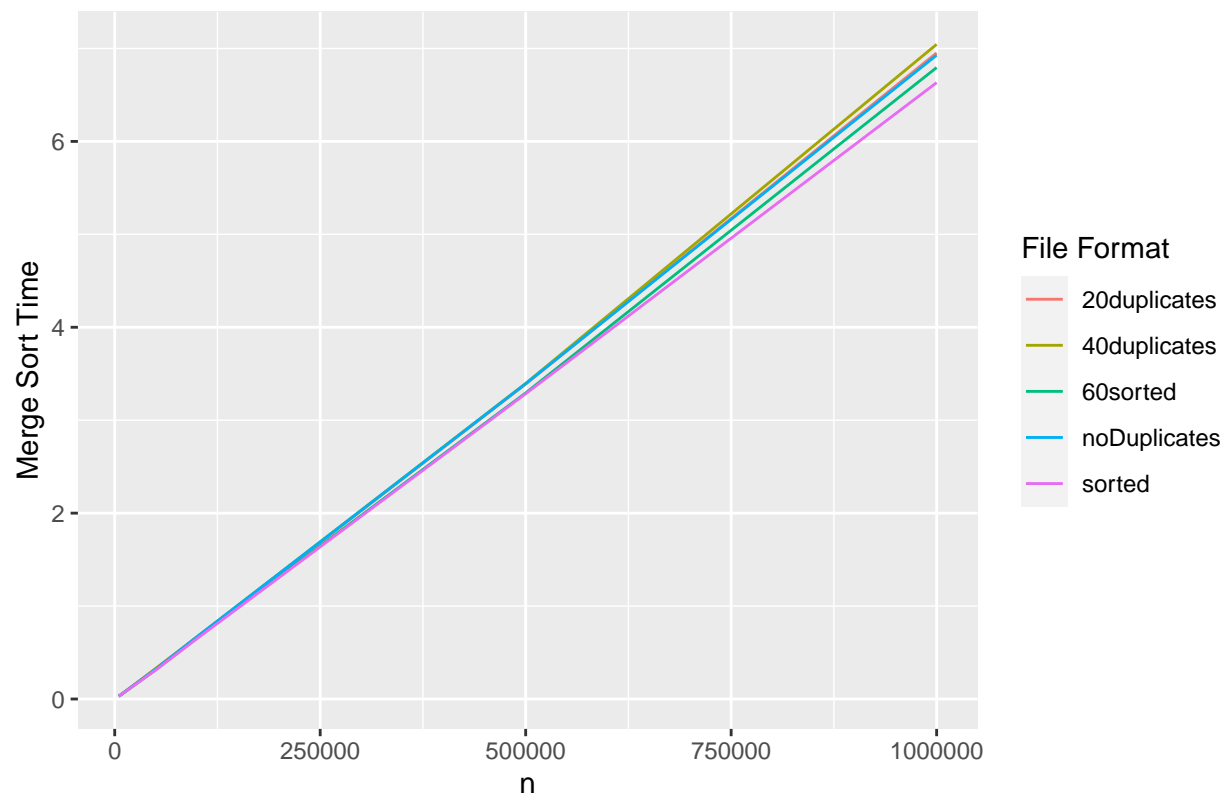
```
## 'geom_smooth()' using formula 'y ~ x'
```


Merge Sort Regression Models By Data Type



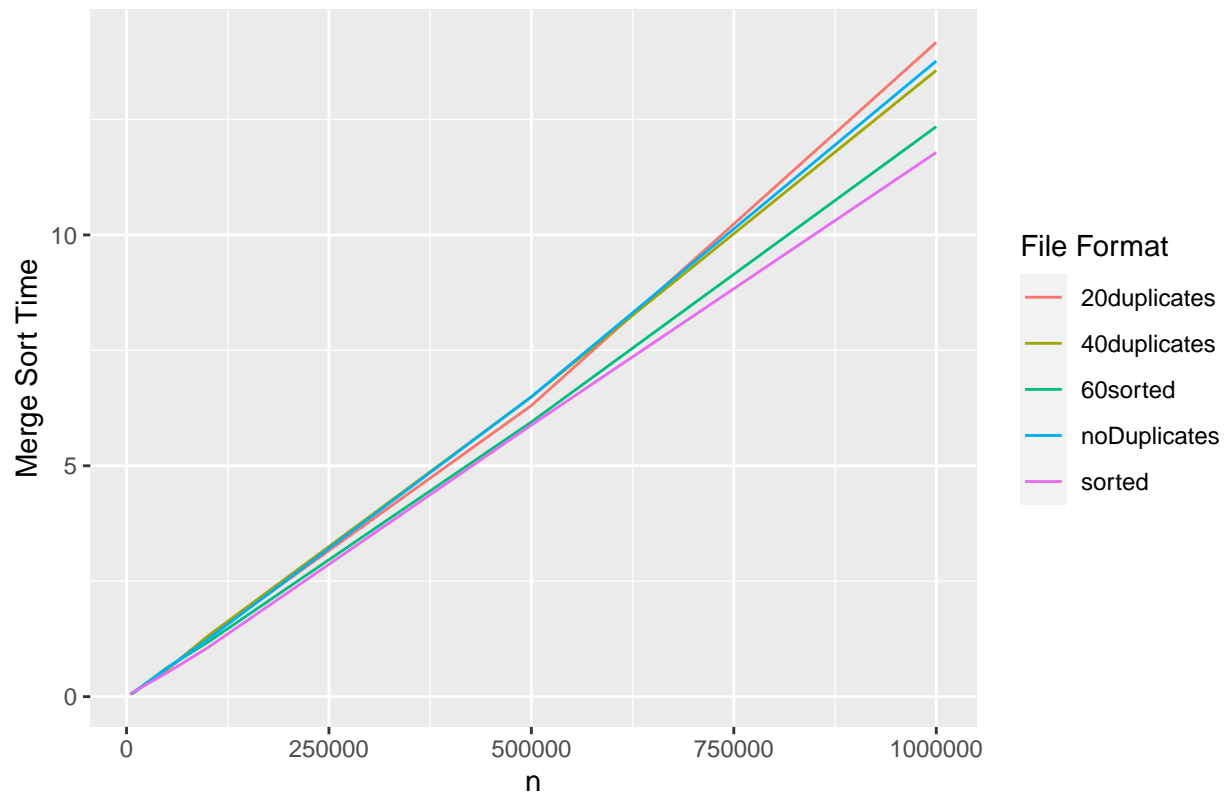
```
mergeInts = subset(mergeTimes, var_type == "int")
ggplot(mergeInts, aes(x = size, y = merge_time, color = format)) +
  geom_line() +
  labs(title = "Merge Sort Time With Integer Data By Data Set Size and File Format", x = "n", y = "Merge")
guides(color = guide_legend(title = "File Format"))
```

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"))
```

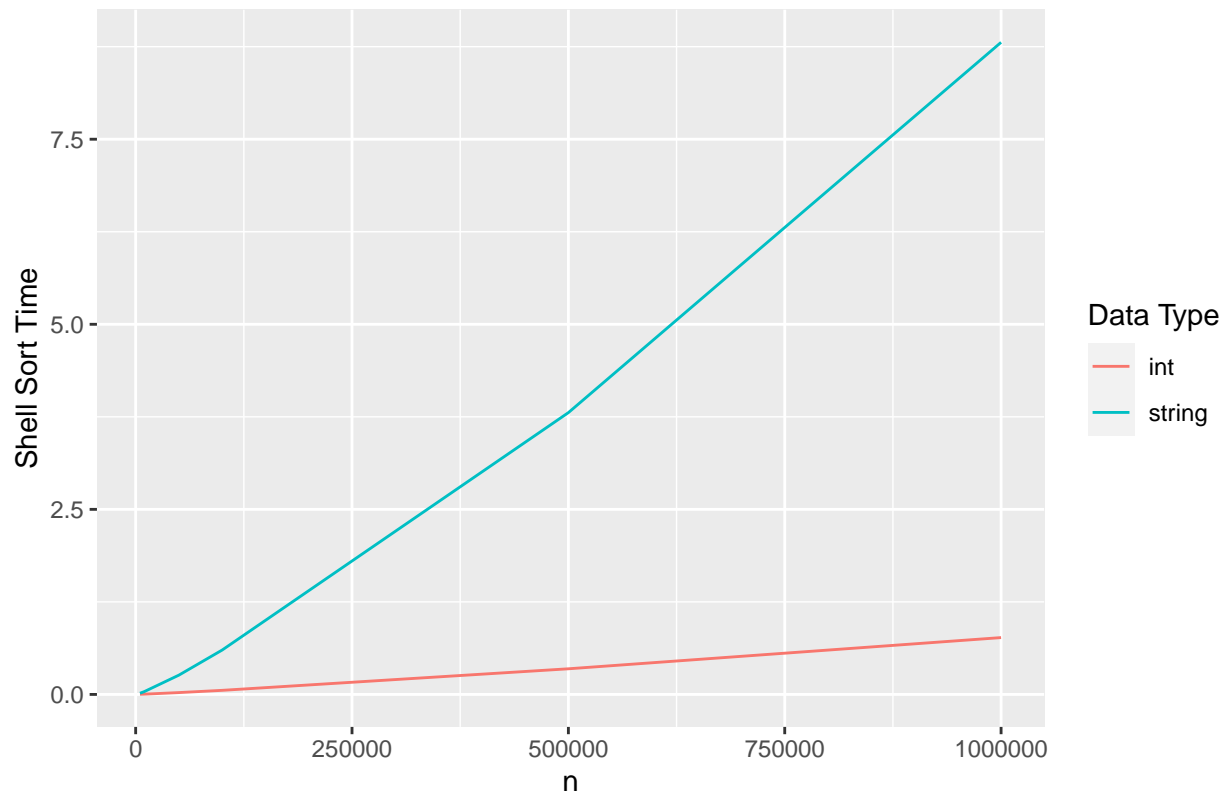
Merge Sort Time With String Data By Data Set Size and File Format



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"))
```

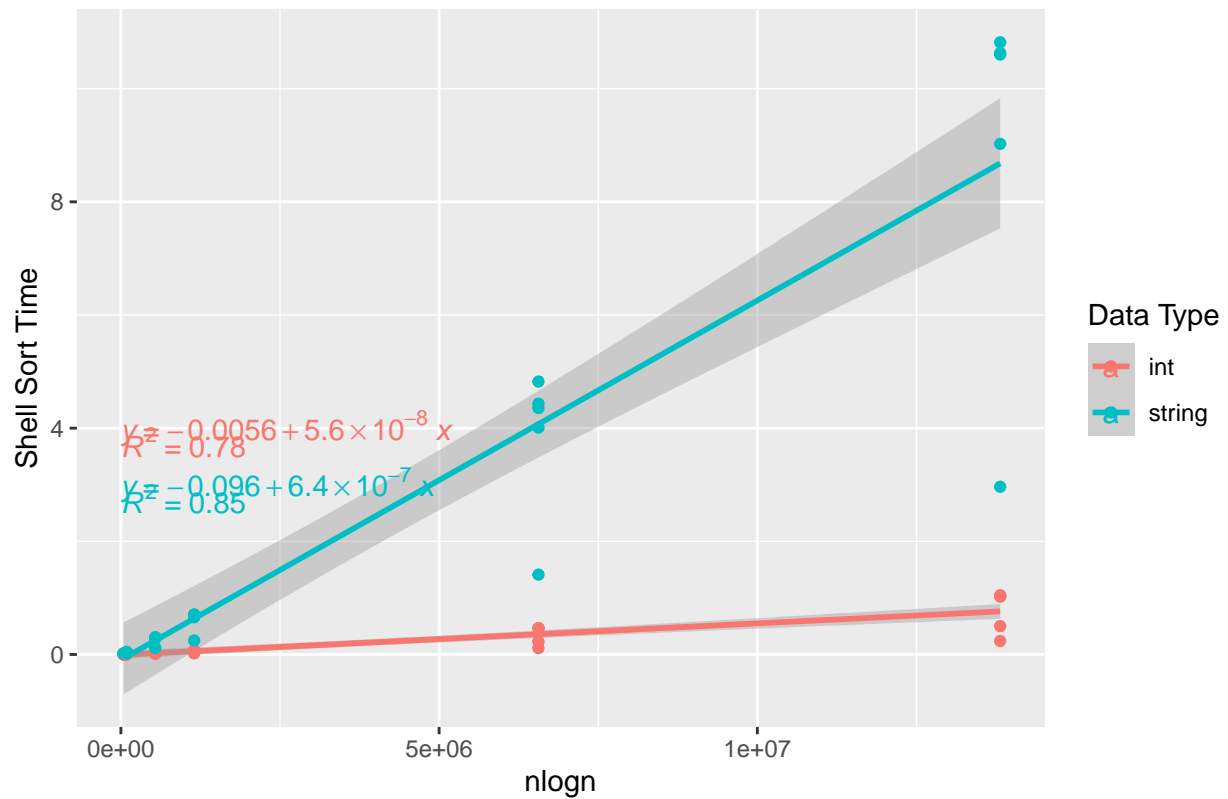
Mean Shell Sort Time By Data Set Size and Data Type



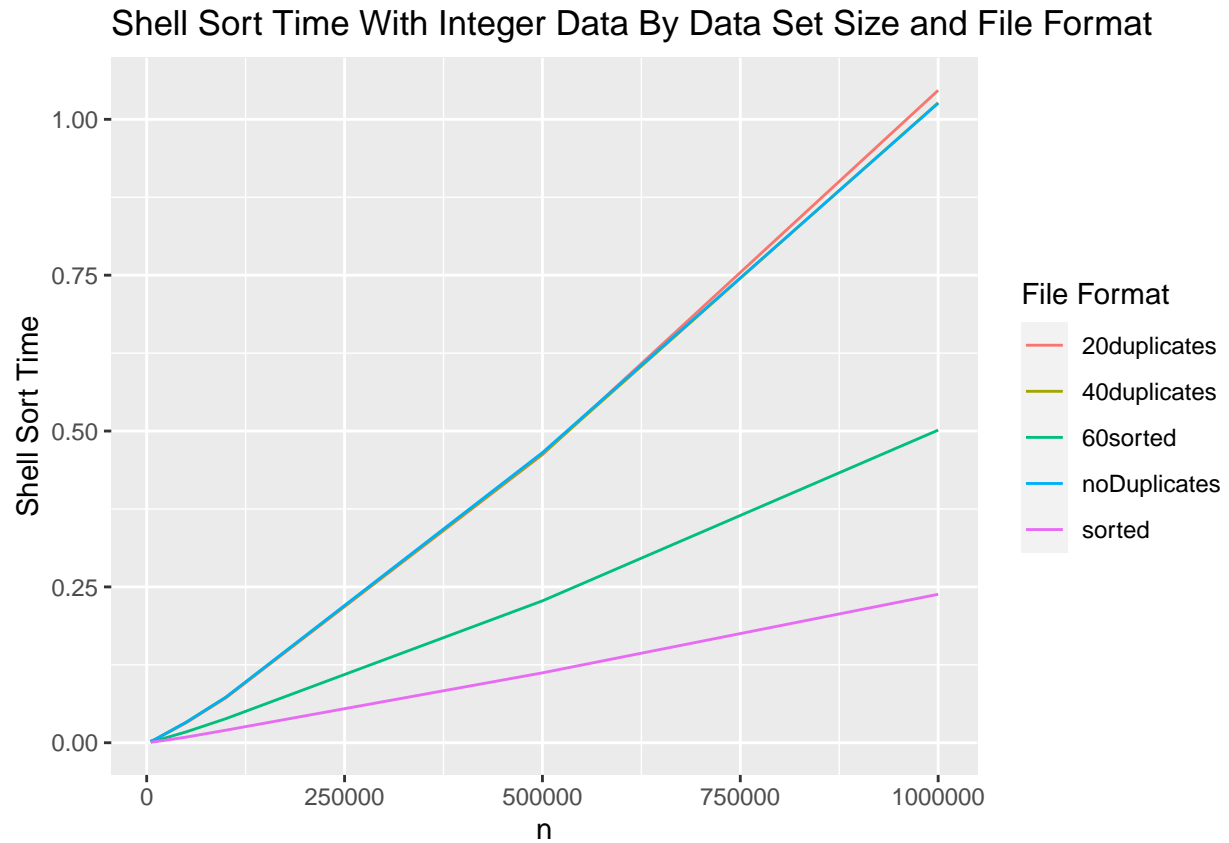
```
ggplot(shellTimes, aes(x = nlogn, y = shell_time, color = var_type)) +  
  labs(title = "Shell Sort Regression Models By Data Type", x = "nlogn", y = "Shell Sort Time") +  
  geom_smooth(method="lm") +  
  geom_point() +  
  stat_regline_equation(label.x=0, label.y=c(4, 3)) +  
  stat_cor(aes(label=..rr.label..),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

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

Shell Sort Regression Models By Data Type

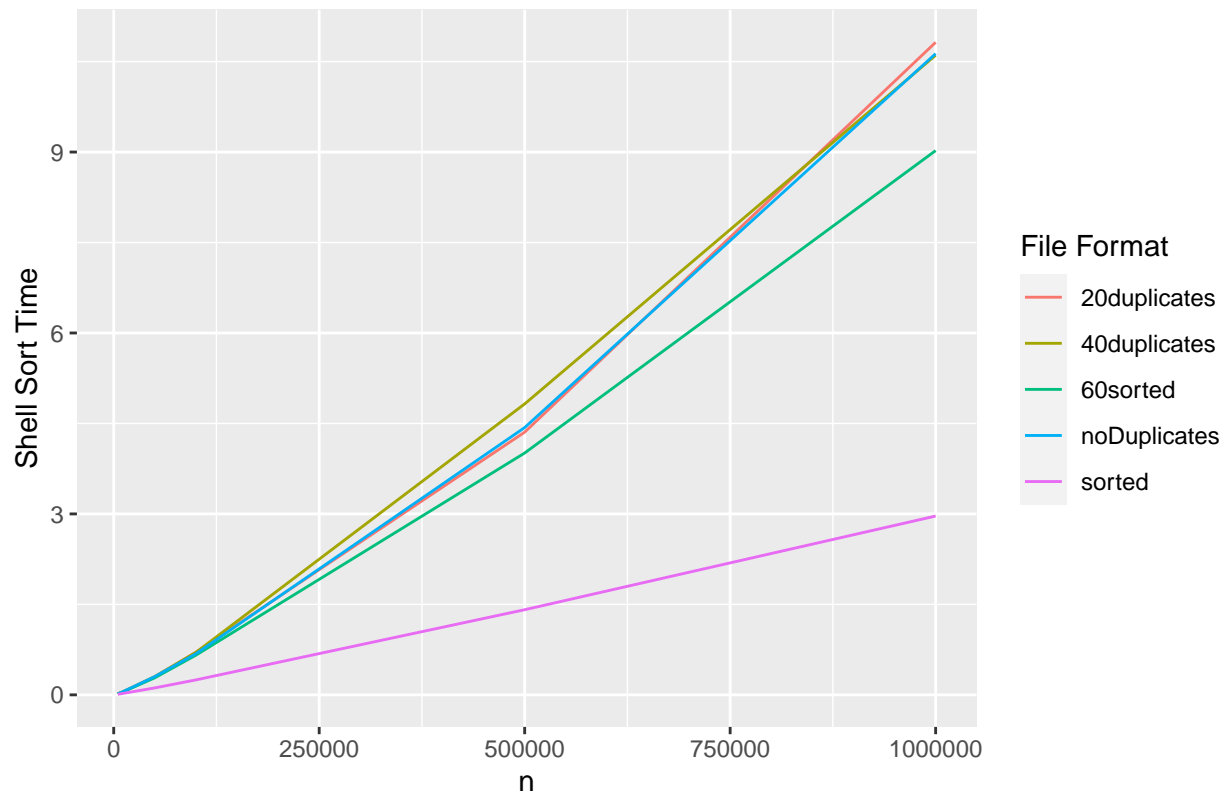


```
shellInts = subset(shellTimes, var_type == "int")
ggplot(shellInts, aes(x = size, y = shell_time, color = format)) +
  geom_line() +
  labs(title = "Shell Sort Time With Integer Data By Data Set Size and File Format", x = "n", y = "Shell")
guides(color = guide_legend(title = "File Format"))
```



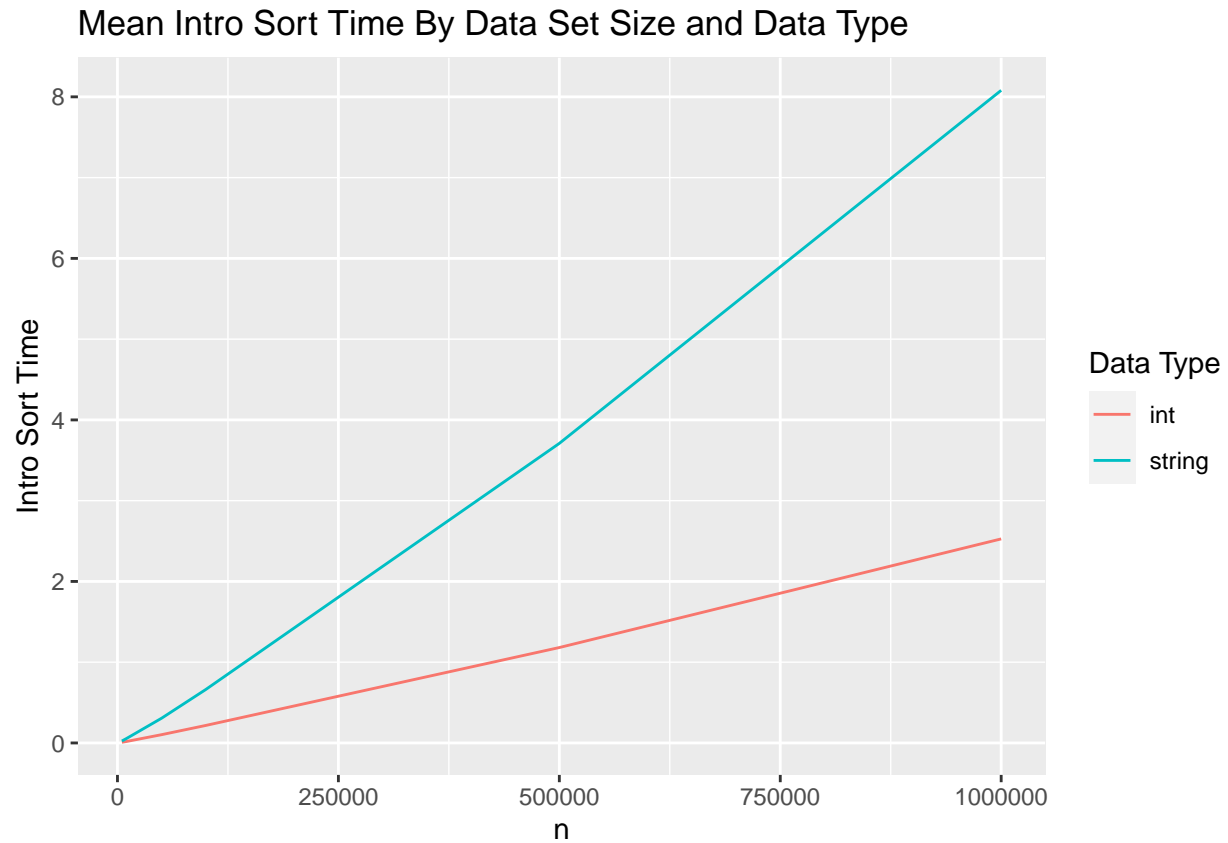
```
shellStrings = subset(shellTimes, var_type == "string")
ggplot(shellStrings, aes(x = size, y = shell_time, color = format)) +
  geom_line() +
  labs(title = "Shell Sort Time With String Data By Data Set Size and File Format", x = "n", y = "Shell")
guides(color = guide_legend(title = "File Format"))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

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



Intro Sort

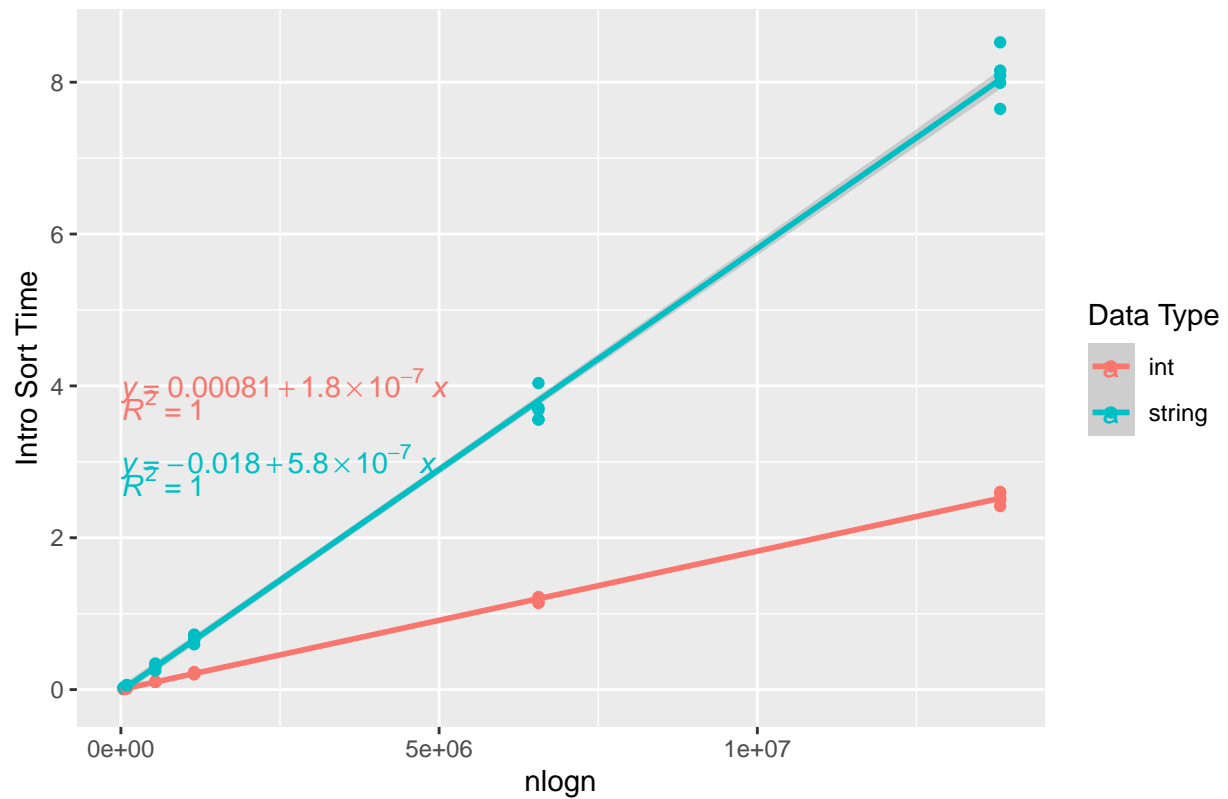
```
introTimes = aggregate(intro_time ~ var_type + size + nlogn + format, data = data, FUN = mean)
introTimes2 = aggregate(intro_time ~ var_type + size + nlogn, data = data, FUN = mean)
ggplot(introTimes2, aes(x = size, y = intro_time, color = var_type)) +
  geom_line() +
  labs(title = "Mean Intro Sort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



```
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"))
```

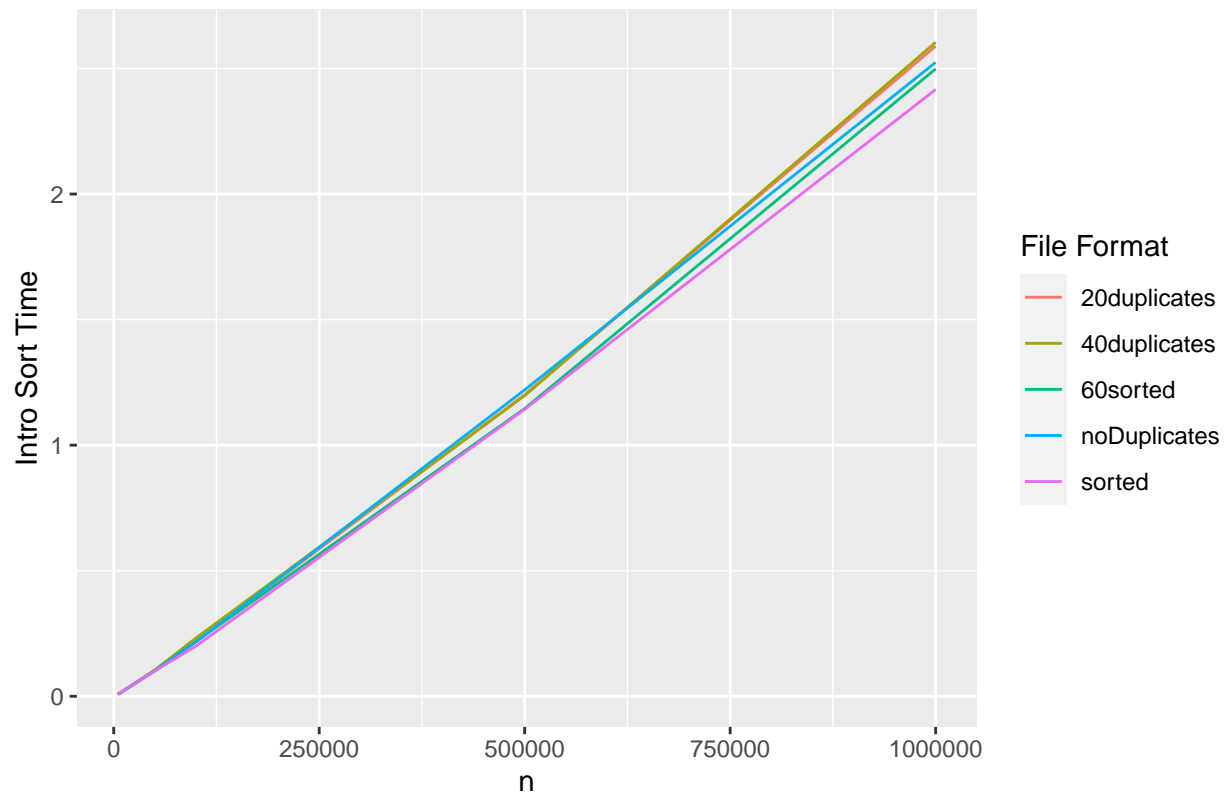
```
## 'geom_smooth()' using formula 'y ~ x'
```


Intro Sort Regression Models By Data Type



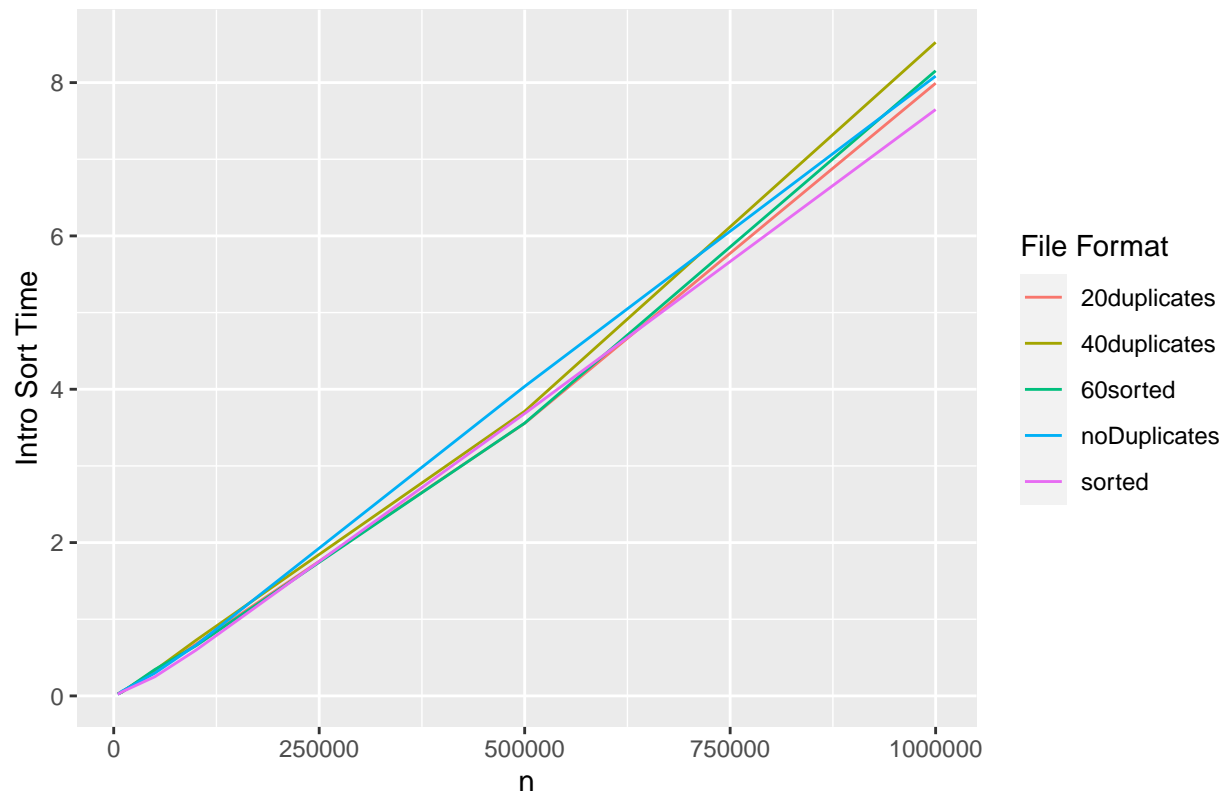
```
introInts = subset(introTimes, var_type == "int")
ggplot(introInts, aes(x = size, y = intro_time, color = format)) +
  geom_line() +
  labs(title = "Intro Sort Time With Integer Data By Data Set Size and File Format", x = "n", y = "Intro")
guides(color = guide_legend(title = "File Format"))
```

Intro Sort Time With Integer Data By Data Set Size and 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"))
```

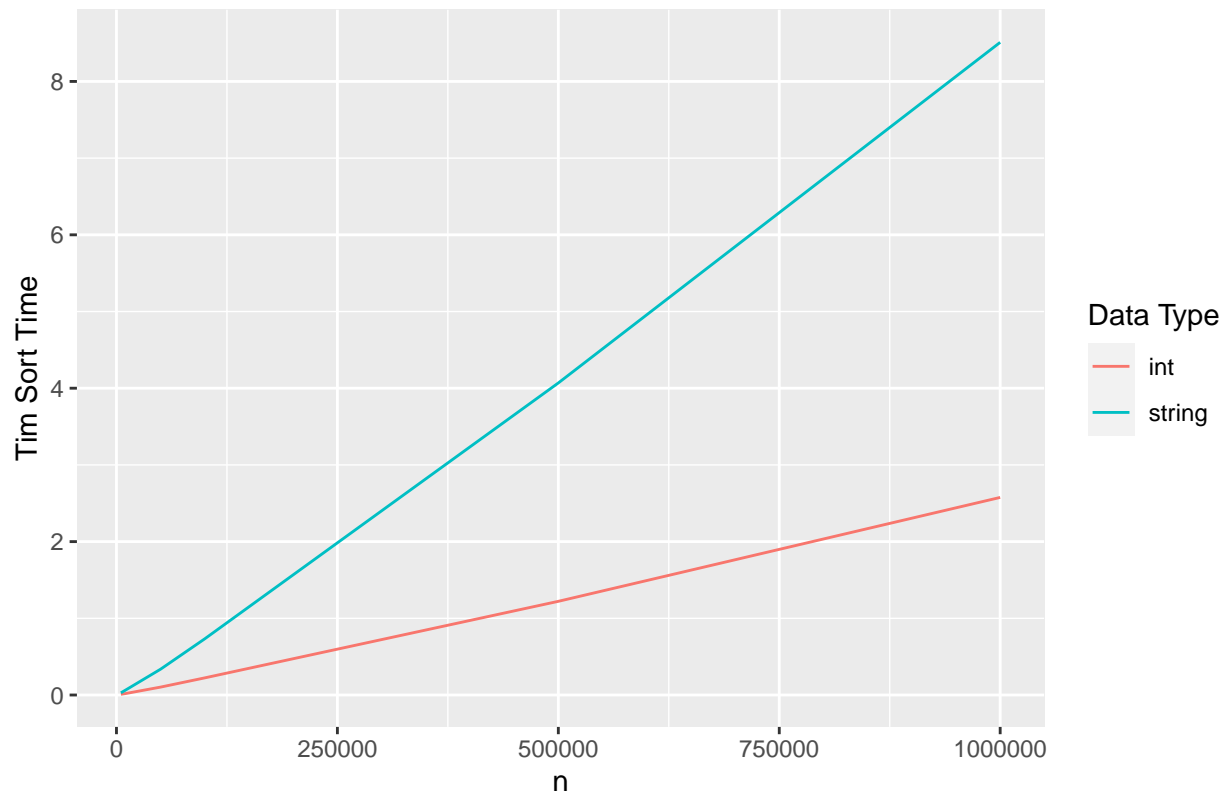
Intro Sort Time With String Data By Data Set Size and File Format



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"))
```

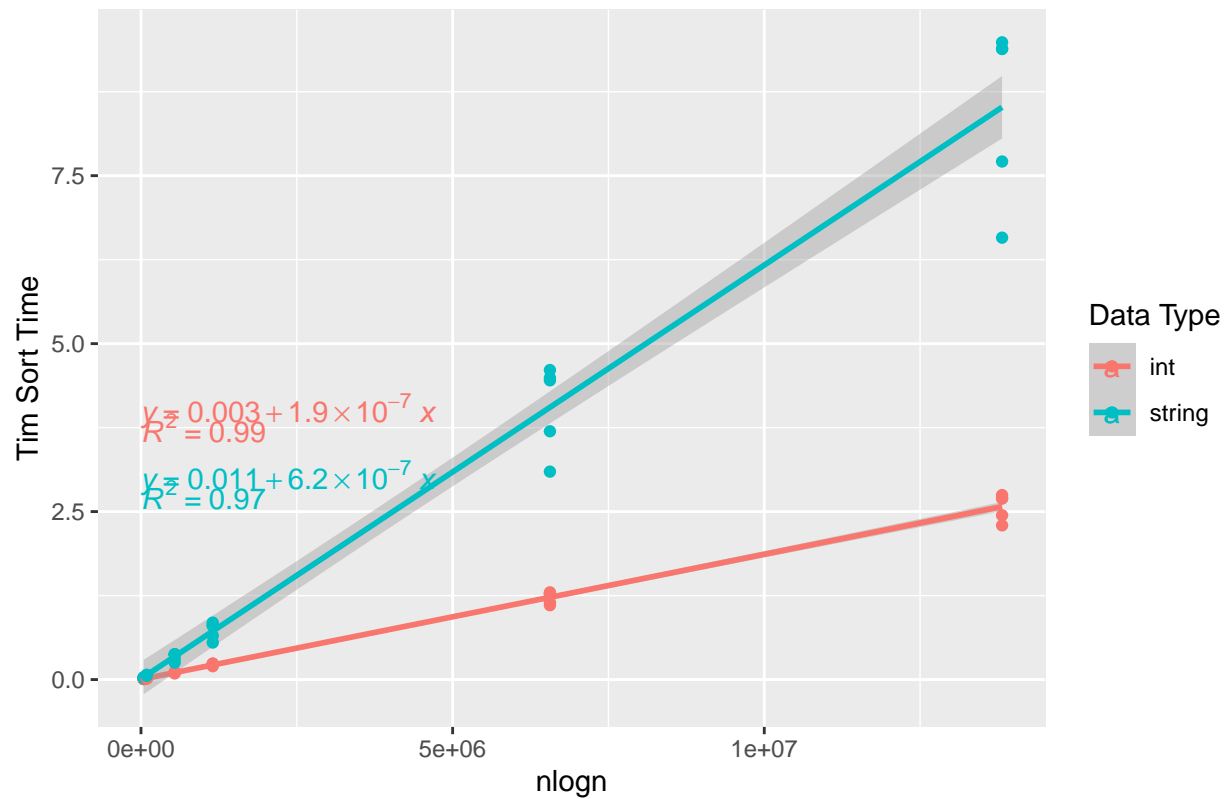
Mean Tim Sort Time By Data Set Size and Data Type



```
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"))
```

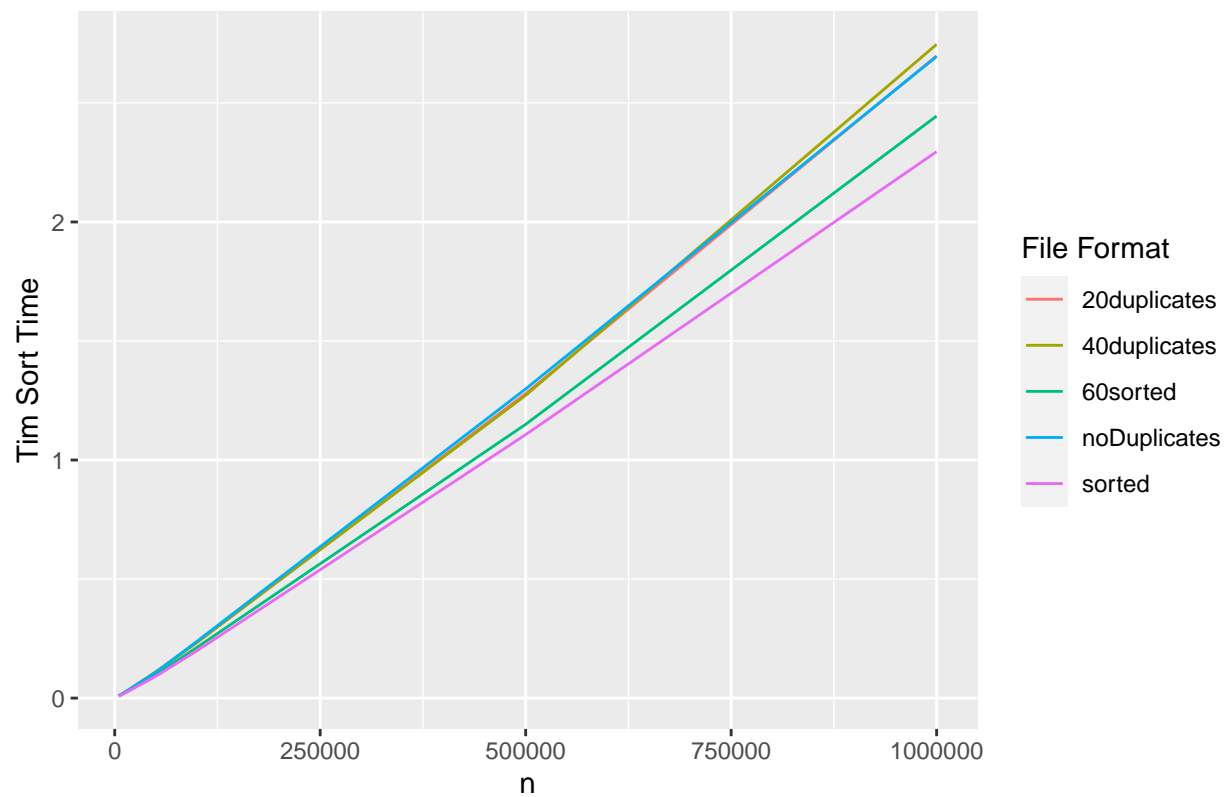
```
## 'geom_smooth()' using formula 'y ~ x'
```

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 Sort Time") +
  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

