

Program 2 Graph Analysis

Ryan Schaefer and Wes Anderson

Create Dataset

```
library(ggplot2)
library(ggpubr)
data = read.csv("RyanDataRun3.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	0.25700200	3.3978000
## 2	int	1000000	40duplicates	N/A	0.52627000	6.9291100
## 3	int	100000	40duplicates	N/A	0.04623580	0.6590070
## 4	int	10000	40duplicates	N/A	0.00395271	0.0615638
## 5	int	50000	sorted	N/A	0.01371850	0.3102540
## 6	int	50000	20duplicates	N/A	0.02247440	0.3215400
## 7	int	5000	noDuplicates	N/A	0.00198078	0.0298307
## 8	int	500000	sorted	N/A	0.18488500	3.2386700
## 9	int	500000	60sorted	N/A	0.21098800	3.2921500
## 10	int	10000	60sorted	N/A	0.00338052	0.0613623
## 11	int	1000000	noDuplicates	N/A	0.51381400	6.9183600
## 12	int	1000000	20duplicates	N/A	0.51835100	6.9223500
## 13	int	50000	noDuplicates	N/A	0.02208670	0.3219480
## 14	int	5000	60sorted	N/A	0.00148984	0.0293975
## 15	int	5000	sorted	N/A	0.00132248	0.0289832
## 16	int	100000	20duplicates	N/A	0.04718360	0.6584070
## 17	int	50000	60sorted	N/A	0.01876540	0.3161380
## 18	int	10000	noDuplicates	N/A	0.00399561	0.0622862
## 19	int	500000	20duplicates	N/A	0.24730500	3.3859100
## 20	int	500000	40duplicates	N/A	0.24848100	3.3811900
## 21	int	1000000	sorted	N/A	0.38055200	6.6186900
## 22	int	5000	20duplicates	N/A	0.00187367	0.0299462
## 23	int	100000	noDuplicates	N/A	0.04601820	0.6629790
## 24	int	50000	40duplicates	N/A	0.02188080	0.3211640
## 25	int	10000	20duplicates	N/A	0.00399606	0.0615672
## 26	int	100000	sorted	N/A	0.02892210	0.6339850
## 27	int	100000	60sorted	N/A	0.03892120	0.6429370
## 28	int	10000	sorted	N/A	0.00283085	0.0597261
## 29	int	5000	40duplicates	N/A	0.00187732	0.0298228
## 30	int	1000000	60sorted	N/A	0.40723100	6.7275800
## 31	string	50000	sorted	N/A	0.13482900	0.5093520
## 32	string	500000	20duplicates	N/A	1.84025000	6.2965700
## 33	string	50000	20duplicates	N/A	0.14254000	0.5704810

## 34	string	10000	40duplicates	N/A	0.02423410	0.1051310
## 35	string	10000	60sorted	N/A	0.02383680	0.0999078
## 36	string	100000	sorted	N/A	0.28107000	1.0535600
## 37	string	5000	40duplicates	N/A	0.01101110	0.0504176
## 38	string	500000	60sorted	N/A	1.71244000	5.8574100
## 39	string	50000	noDuplicates	N/A	0.14172300	0.6129040
## 40	string	500000	40duplicates	N/A	1.85689000	6.3491600
## 41	string	5000	20duplicates	N/A	0.01102970	0.0504444
## 42	string	100000	noDuplicates	N/A	0.32066000	1.1766600
## 43	string	5000	noDuplicates	N/A	0.01137380	0.0503983
## 44	string	100000	60sorted	N/A	0.30755200	1.1195800
## 45	string	1000000	20duplicates	N/A	3.77451000	13.2816000
## 46	string	10000	noDuplicates	N/A	0.02747040	0.1058840
## 47	string	1000000	noDuplicates	N/A	3.78058000	13.3243000
## 48	string	1000000	sorted	N/A	3.72907000	11.5698000
## 49	string	500000	noDuplicates	N/A	1.78635000	6.5589800
## 50	string	100000	40duplicates	N/A	0.29810800	1.2731000
## 51	string	5000	60sorted	N/A	0.01073290	0.0476549
## 52	string	1000000	60sorted	N/A	3.65559000	12.4722000
## 53	string	5000	sorted	N/A	0.01103610	0.0497344
## 54	string	100000	20duplicates	N/A	0.30644300	1.2356900
## 55	string	10000	20duplicates	N/A	0.02330170	0.1063390
## 56	string	10000	sorted	N/A	0.02370990	0.0981268
## 57	string	500000	sorted	N/A	1.67760000	5.6440300
## 58	string	50000	40duplicates	N/A	0.14571700	0.5895810
## 59	string	50000	60sorted	N/A	0.14084100	0.5881920
## 60	string	1000000	40duplicates	N/A	3.82892000	13.4541000
##	shell_time	intro_time	tim_time	n2	nlogn	
## 1	4.64797e-01	1.22404000	1.28858000	2.5e+11	6561181.69	
## 2	1.02046e+00	2.54508000	2.71308000	1.0e+12	13815510.56	
## 3	7.31882e-02	0.21697600	0.23012100	1.0e+10	1151292.55	
## 4	4.72400e-03	0.01698230	0.01913080	1.0e+08	92103.40	
## 5	9.04236e-03	0.09269730	0.09315070	2.5e+09	540988.91	
## 6	3.20885e-02	0.10103100	0.10867200	2.5e+09	540988.91	
## 7	2.18139e-03	0.00817862	0.00896229	2.5e+07	42585.97	
## 8	1.10429e-01	1.15707000	1.07377000	2.5e+11	6561181.69	
## 9	2.26999e-01	1.14206000	1.15177000	2.5e+11	6561181.69	
## 10	2.72057e-03	0.01645490	0.01732560	1.0e+08	92103.40	
## 11	1.02338e+00	2.49790000	2.70300000	1.0e+12	13815510.56	
## 12	1.04278e+00	2.72222000	2.69620000	1.0e+12	13815510.56	
## 13	3.29464e-02	0.10476800	0.10943600	2.5e+09	540988.91	
## 14	1.17768e-03	0.00744205	0.00806057	2.5e+07	42585.97	
## 15	7.07377e-04	0.00743803	0.00754819	2.5e+07	42585.97	
## 16	7.19400e-02	0.21797700	0.22930400	1.0e+10	1151292.55	
## 17	1.75556e-02	0.09177400	0.09824270	2.5e+09	540988.91	
## 18	4.90775e-03	0.01722850	0.01903180	1.0e+08	92103.40	
## 19	4.61394e-01	1.18024000	1.26941000	2.5e+11	6561181.69	
## 20	4.60087e-01	1.21834000	1.26709000	2.5e+11	6561181.69	
## 21	2.34770e-01	2.53920000	2.28589000	1.0e+12	13815510.56	
## 22	2.11779e-03	0.00781291	0.00892561	2.5e+07	42585.97	
## 23	7.29862e-02	0.22077500	0.23230200	1.0e+10	1151292.55	
## 24	3.25677e-02	0.09990720	0.10796000	2.5e+09	540988.91	
## 25	4.85882e-03	0.01739950	0.01910060	1.0e+08	92103.40	
## 26	1.94055e-02	0.19510300	0.19644600	1.0e+10	1151292.55	

```
## 27 3.75956e-02 0.20734500 0.20858000 1.0e+10 1151292.55
## 28 1.54822e-03 0.01635800 0.01623130 1.0e+08 92103.40
## 29 2.17045e-03 0.00795529 0.00890045 2.5e+07 42585.97
## 30 5.01328e-01 2.50507000 2.43495000 1.0e+12 13815510.56
## 31 1.15347e-01 0.27773200 0.25253600 2.5e+09 540988.91
## 32 4.33526e+00 3.47743000 4.40902000 2.5e+11 6561181.69
## 33 3.00086e-01 0.28446700 0.36830700 2.5e+09 540988.91
## 34 4.24574e-02 0.04790630 0.06551850 1.0e+08 92103.40
## 35 4.10682e-02 0.04753470 0.05116320 1.0e+08 92103.40
## 36 2.46325e-01 0.57558100 0.52744200 1.0e+10 1151292.55
## 37 1.82465e-02 0.02213030 0.02991680 2.5e+07 42585.97
## 38 3.98478e+00 3.71094000 3.62276000 2.5e+11 6561181.69
## 39 2.98531e-01 0.35725100 0.36991600 2.5e+09 540988.91
## 40 4.77025e+00 3.66558000 4.40671000 2.5e+11 6561181.69
## 41 1.82394e-02 0.02204000 0.02974870 2.5e+07 42585.97
## 42 6.78132e-01 0.67223200 0.77817600 1.0e+10 1151292.55
## 43 1.84962e-02 0.02451590 0.02979000 2.5e+07 42585.97
## 44 6.52926e-01 0.63236900 0.63454300 1.0e+10 1151292.55
## 45 1.06763e+01 7.59610000 9.35646000 1.0e+12 13815510.56
## 46 4.40530e-02 0.06931990 0.06671720 1.0e+08 92103.40
## 47 1.03187e+01 7.74702000 9.30548000 1.0e+12 13815510.56
## 48 2.95884e+00 7.19609000 6.46131000 1.0e+12 13815510.56
## 49 4.39294e+00 3.59125000 4.43271000 2.5e+11 6561181.69
## 50 6.95108e-01 0.74360700 0.78636300 1.0e+10 1151292.55
## 51 1.68351e-02 0.02630480 0.02377700 2.5e+07 42585.97
## 52 8.93375e+00 7.87848000 7.64056000 1.0e+12 13815510.56
## 53 9.03418e-03 0.02781430 0.01947900 2.5e+07 42585.97
## 54 7.01887e-01 0.66887900 0.80186200 1.0e+10 1151292.55
## 55 4.29113e-02 0.04760460 0.06753150 1.0e+08 92103.40
## 56 1.97929e-02 0.04519550 0.04238430 1.0e+08 92103.40
## 57 1.39664e+00 3.76872000 3.01407000 2.5e+11 6561181.69
## 58 3.00959e-01 0.29102400 0.36662900 2.5e+09 540988.91
## 59 2.96464e-01 0.31896500 0.29550100 2.5e+09 540988.91
## 60 1.02428e+01 8.09067000 9.30958000 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"))
```

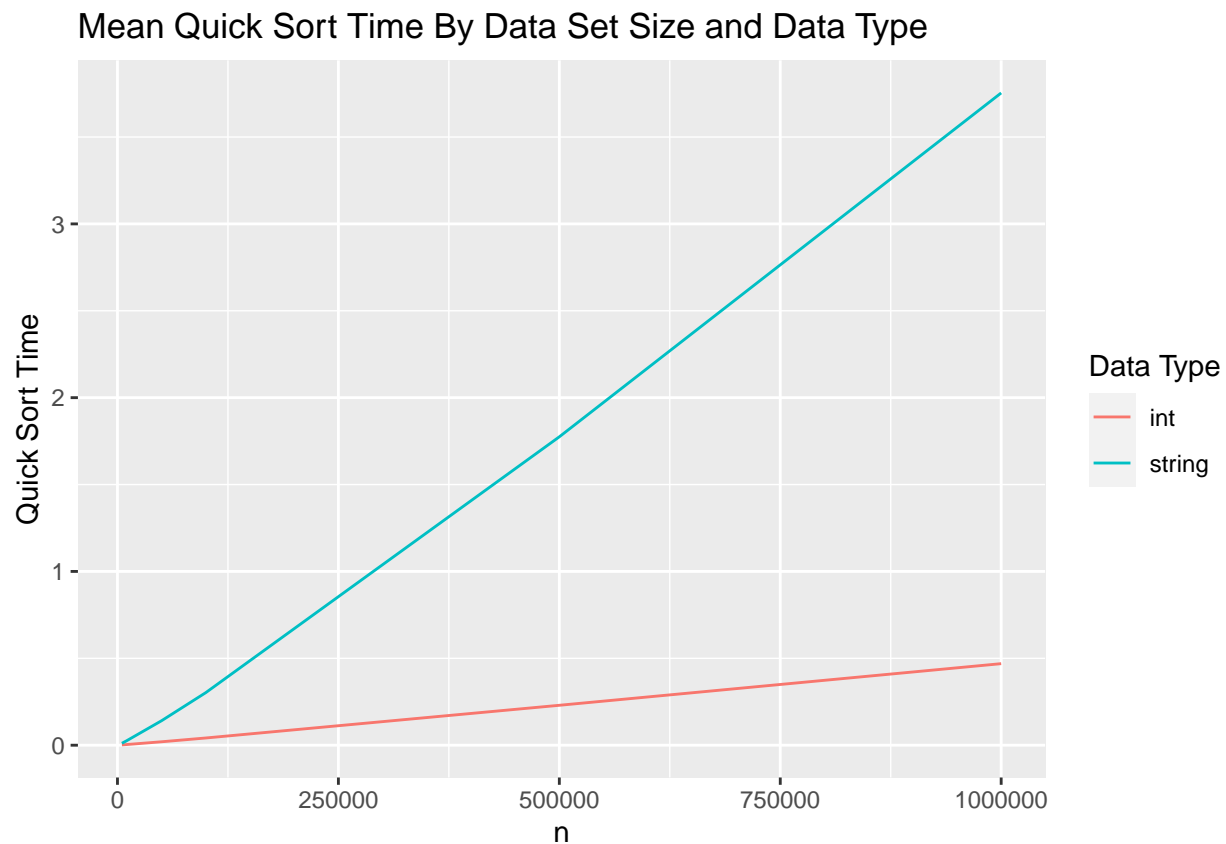
```
# ggplot(insertionTimes, aes(x = size, y = insertion_time, color = var_type)) +
#   labs(title = "Insertion Sort Regression Models By Data Type", x = "n^2", y = "Insertion Sort Time")
#   geom_smooth(method="lm") +
#   geom_point() +
#   stat_regline_equation(label.x=0, label.y=c(9000, 6000)) +
#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

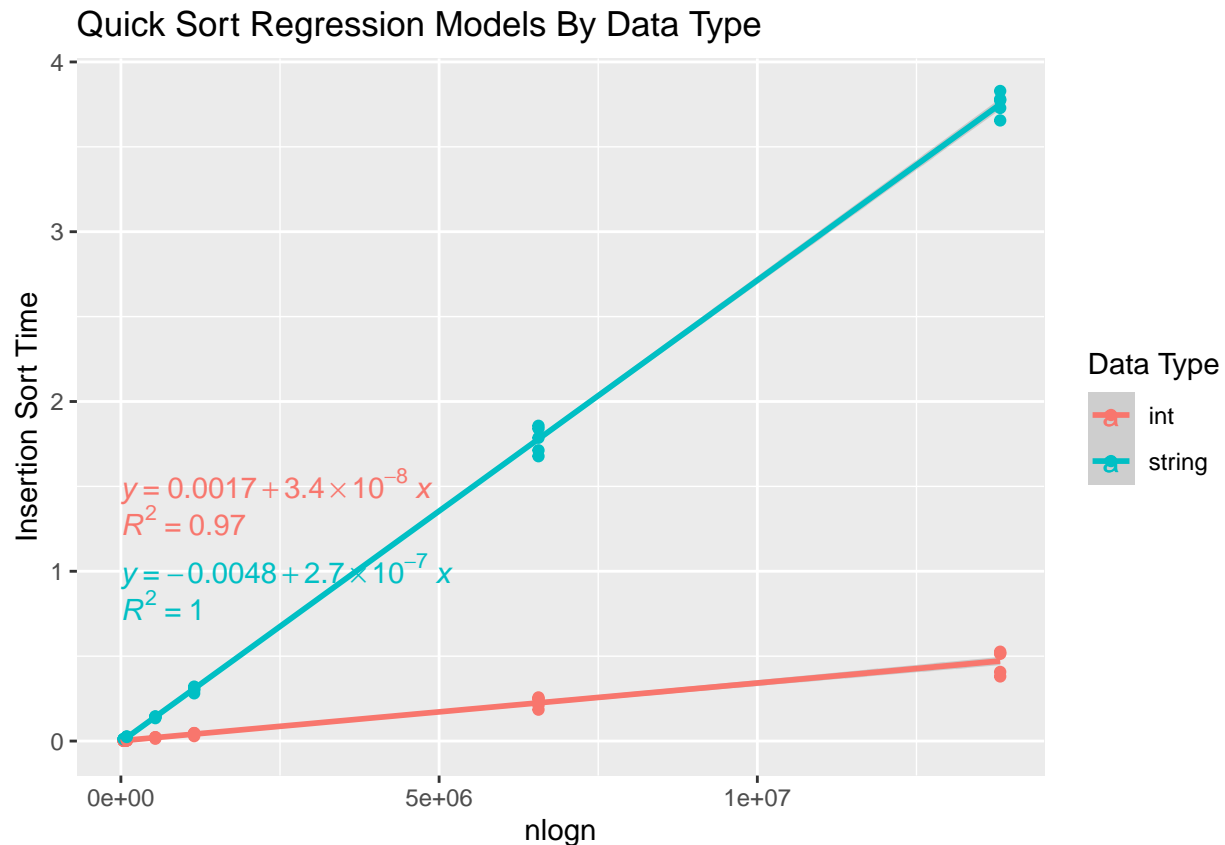
Quick Sort

```
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"))
```



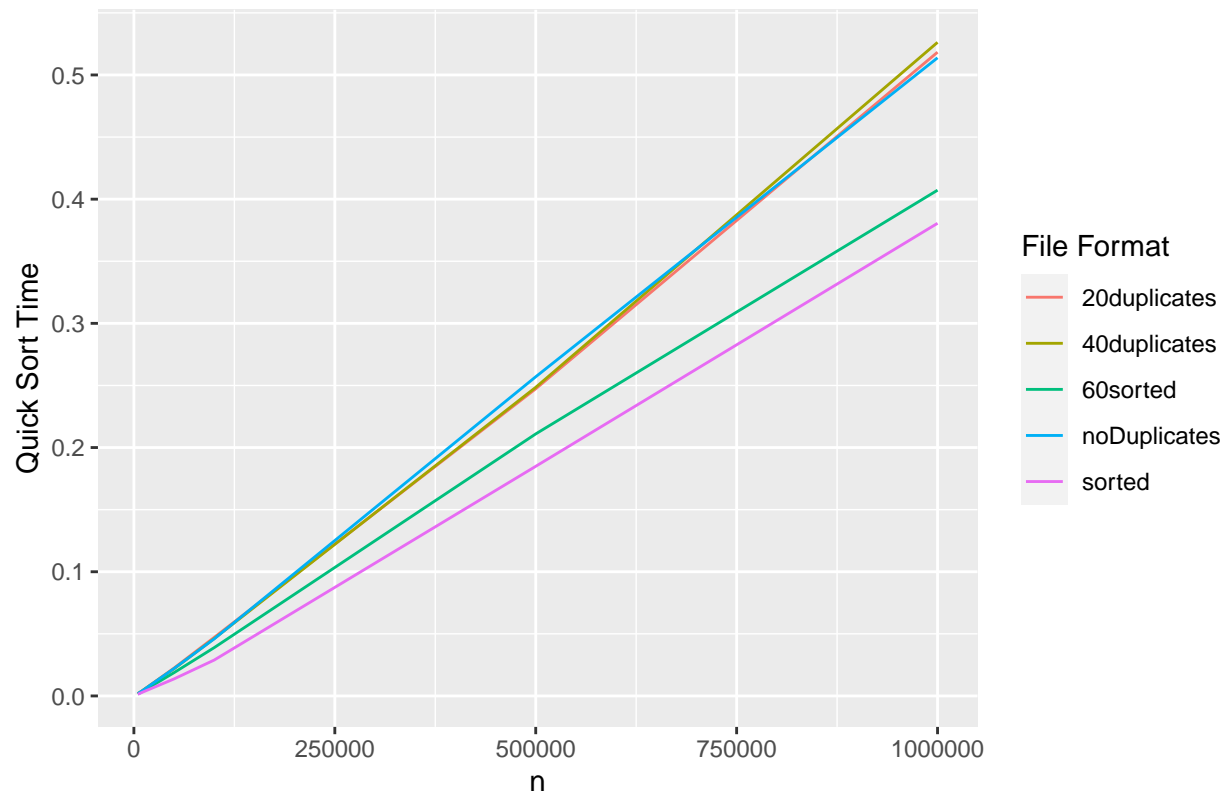
```
ggplot(quickTimes, aes(x = nlogn, y = quick_time, color = var_type)) +
  labs(title = "Quick Sort Regression Models By Data Type", x = "nlogn", y = "Insertion Sort Time") +
  geom_smooth(method="lm") +
  geom_point() +
  stat_regline_equation(label.x=0, label.y=c(1.5, 1)) +
  stat_cor(aes(label=..rr.label..),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

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

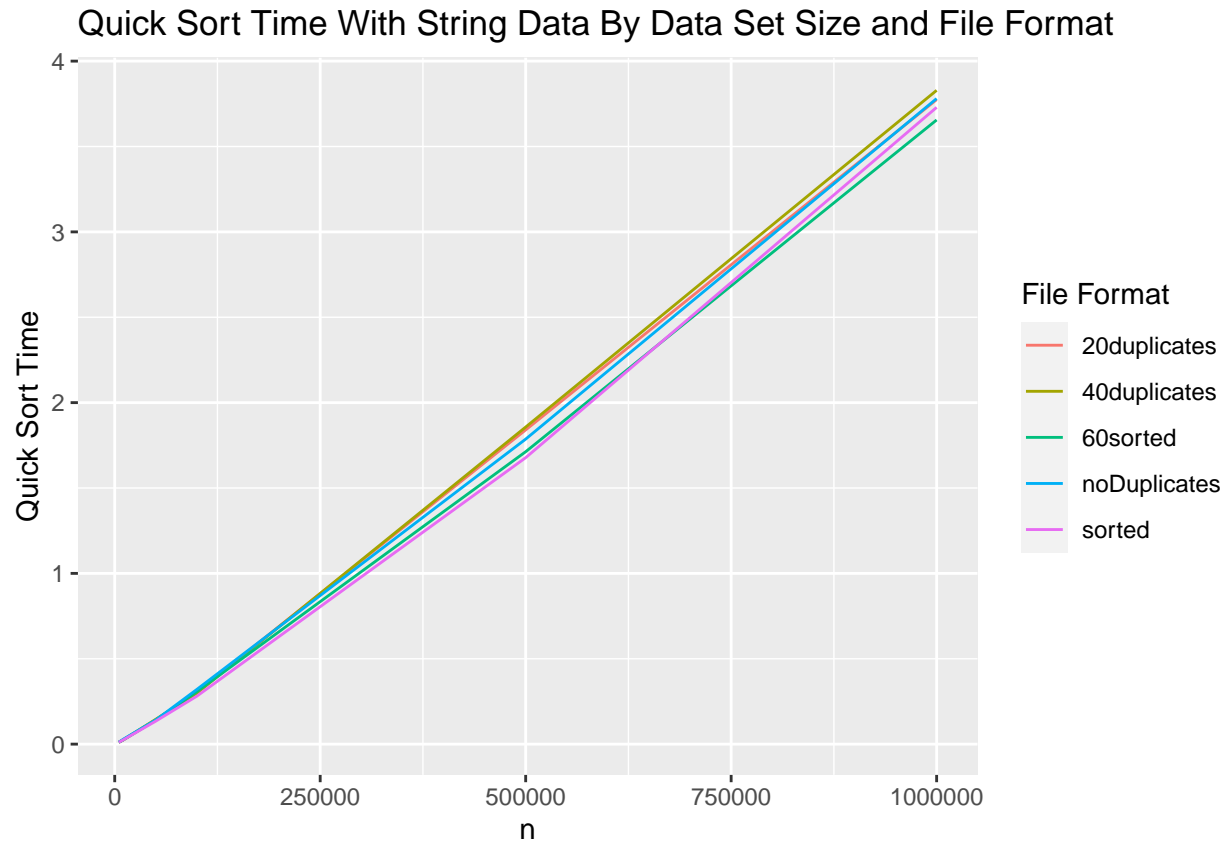


```
quickInts = subset(quickTimes, var_type == "int")
ggplot(quickInts, aes(x = size, y = quick_time, color = format)) +
  geom_line() +
  labs(title = "Quick Sort Time With Integer Data By Data Set Size and File Format", x = "n", y = "Quick")
  guides(color = guide_legend(title = "File Format"))
```

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



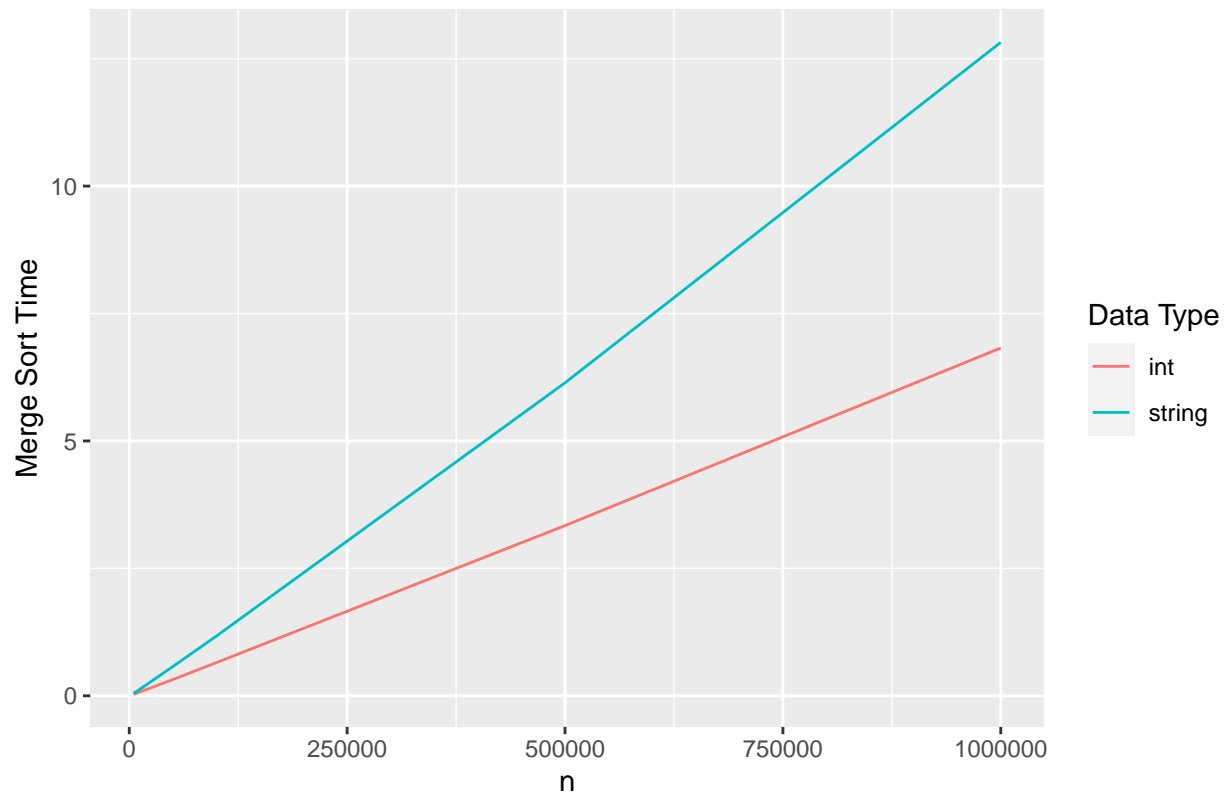
```
quickStrings = subset(quickTimes, var_type == "string")
ggplot(quickStrings, aes(x = size, y = quick_time, color = format)) +
  geom_line() +
  labs(title = "Quick Sort Time With String Data By Data Set Size and File Format", x = "n", y = "Quick")
guides(color = guide_legend(title = "File Format"))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



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"))
```

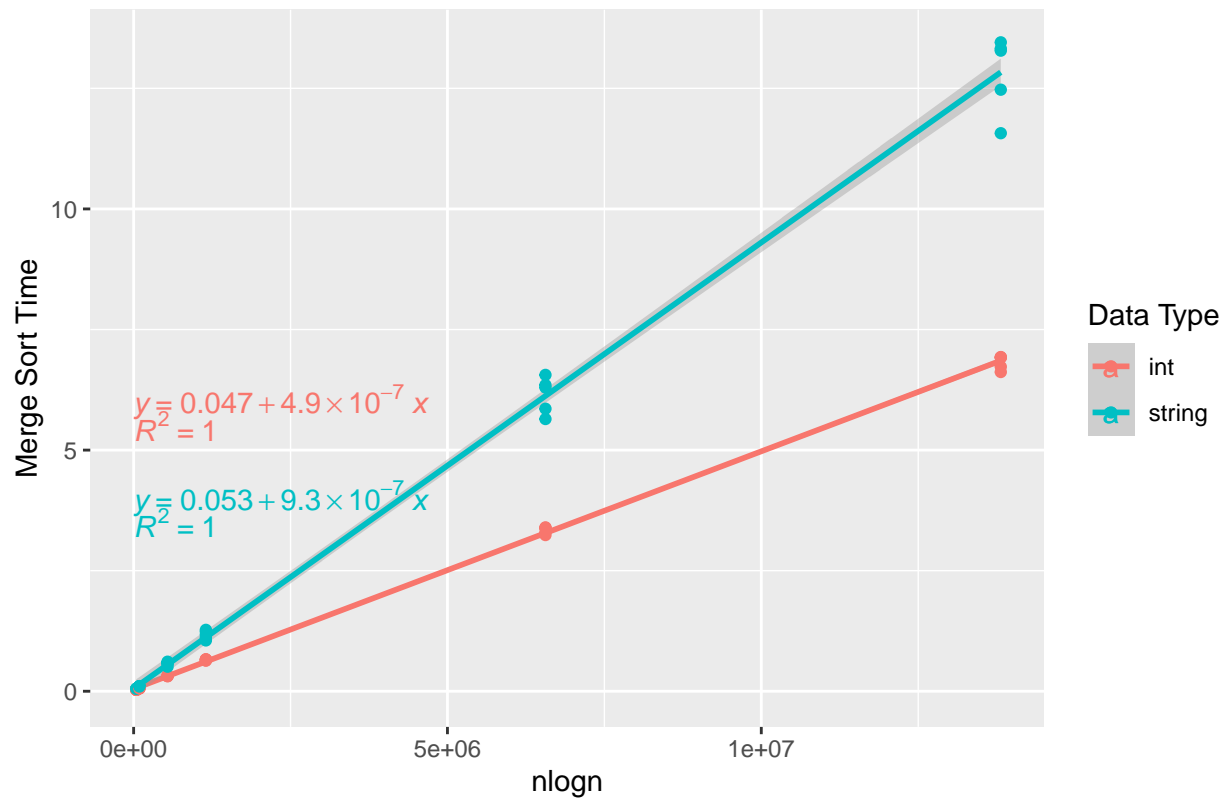
Mean Merge Sort Time By Data Set Size and Data Type



```
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"))
```

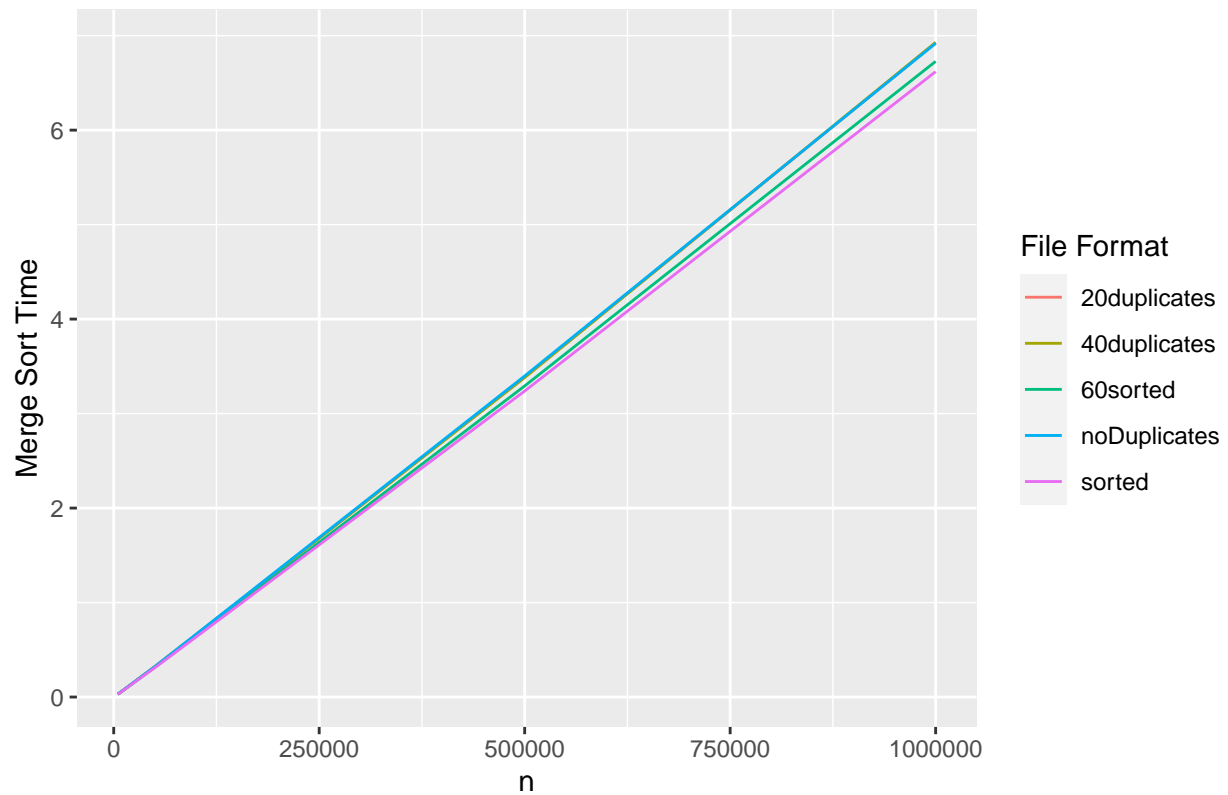
```
## 'geom_smooth()' using formula 'y ~ x'
```


Merge Sort Regression Models By Data Type



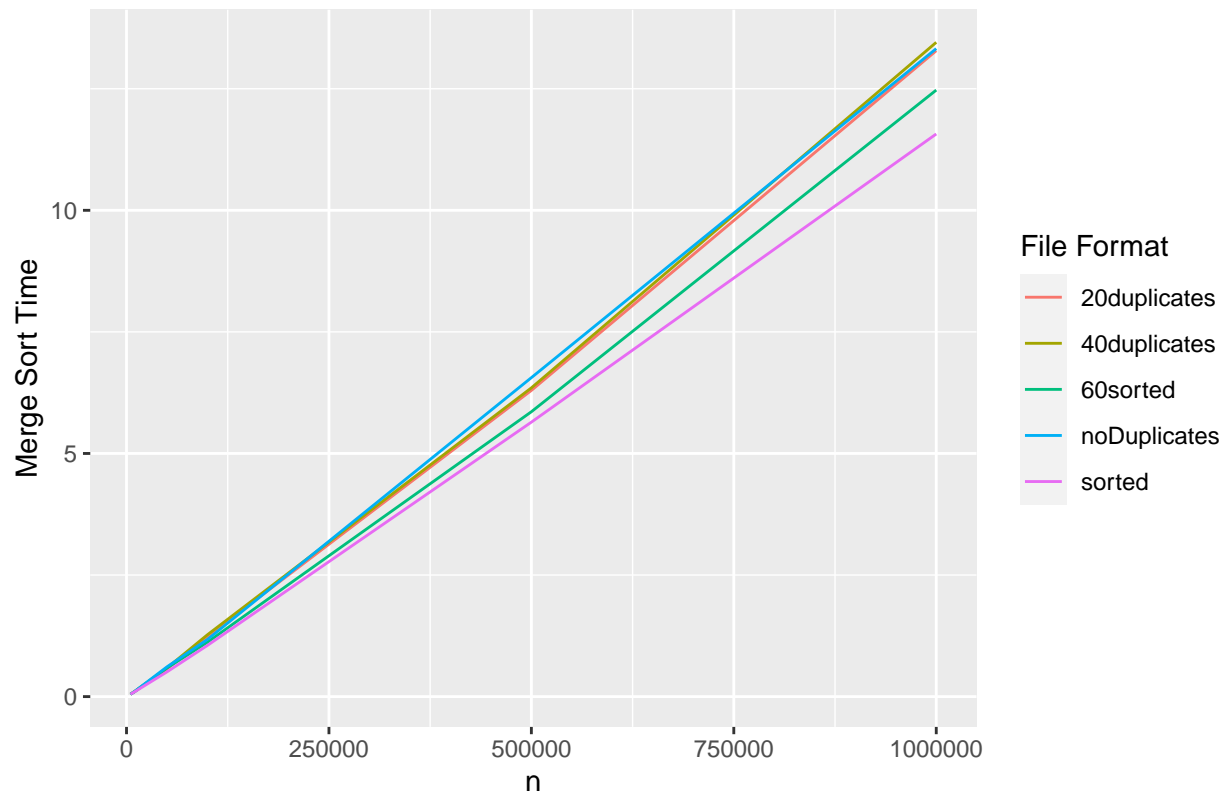
```
mergeInts = subset(mergeTimes, var_type == "int")
ggplot(mergeInts, aes(x = size, y = merge_time, color = format)) +
  geom_line() +
  labs(title = "Merge Sort Time With Integer Data By Data Set Size and File Format", x = "n", y = "Merge")
guides(color = guide_legend(title = "File Format"))
```

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"))
```

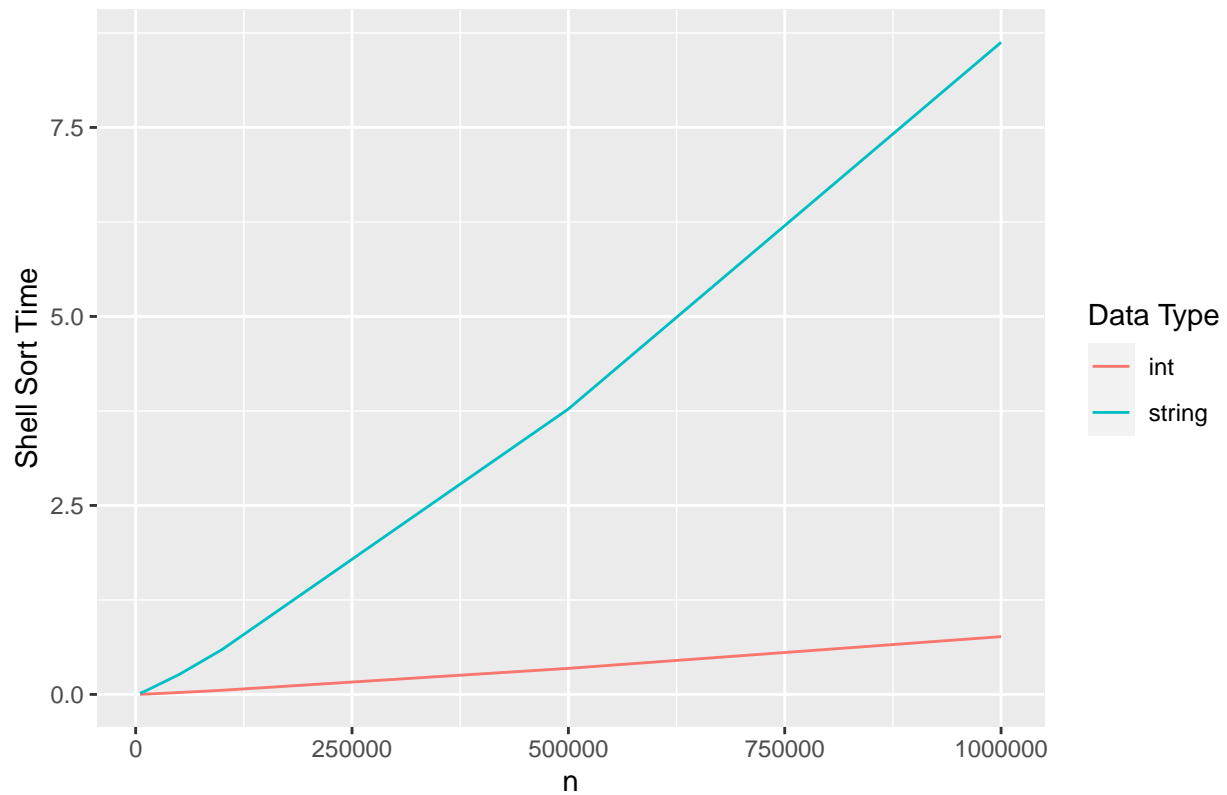
Merge Sort Time With String Data By Data Set Size and File Format



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"))
```

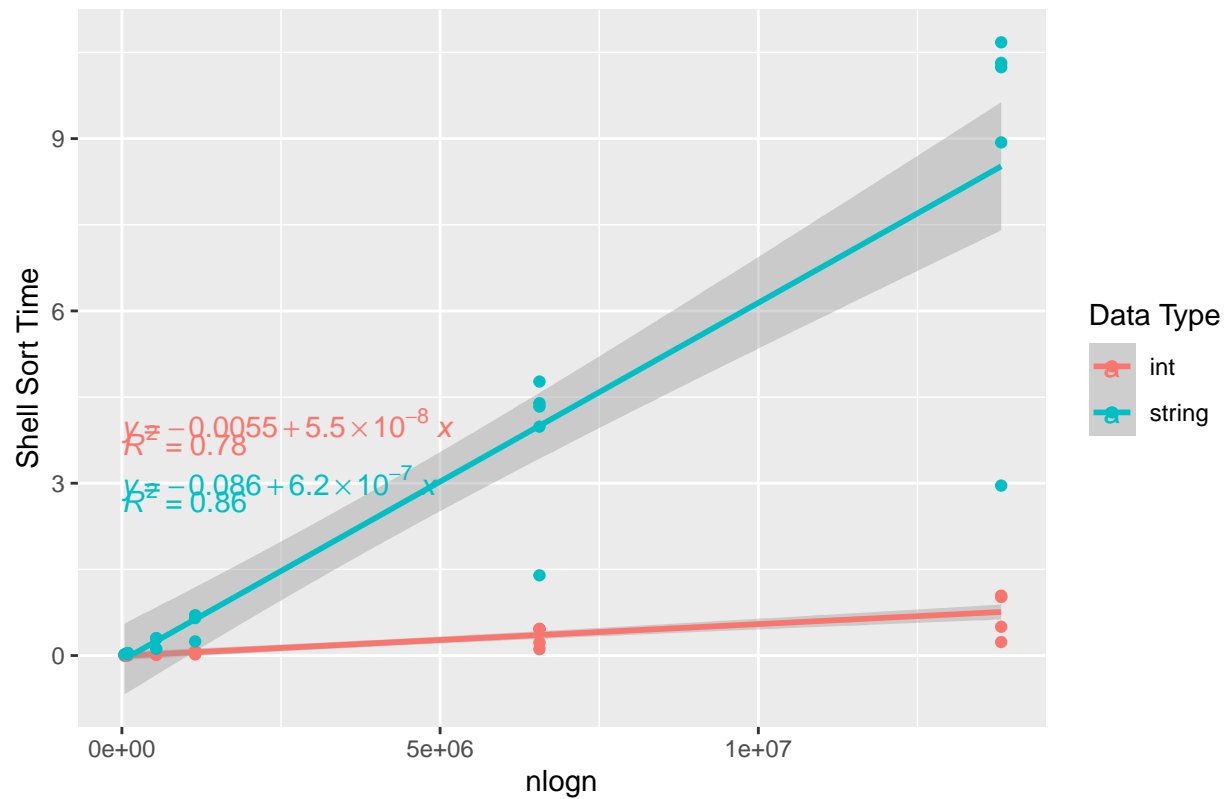
Mean Shell Sort Time By Data Set Size and Data Type



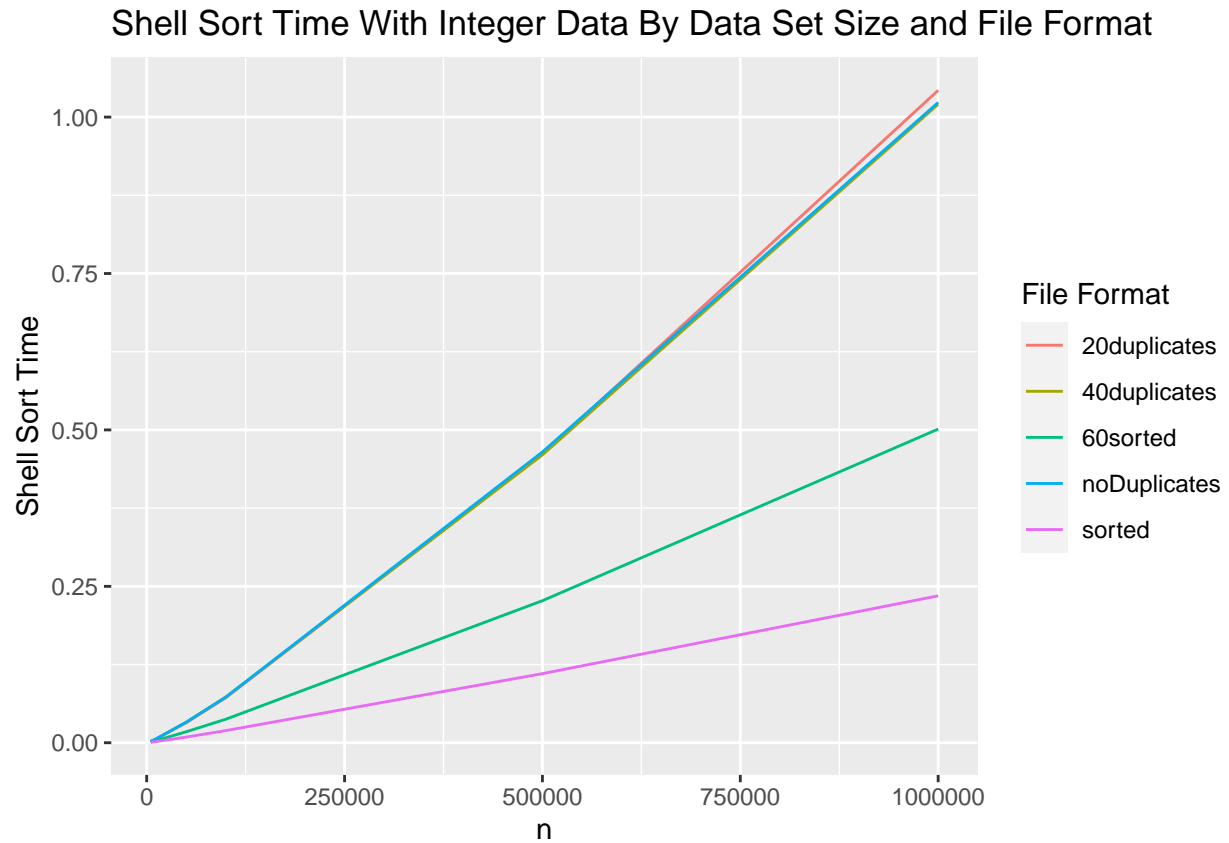
```
ggplot(shellTimes, aes(x = nlogn, y = shell_time, color = var_type)) +  
  labs(title = "Shell Sort Regression Models By Data Type", x = "nlogn", y = "Shell Sort Time") +  
  geom_smooth(method="lm") +  
  geom_point() +  
  stat_regline_equation(label.x=0, label.y=c(4, 3)) +  
  stat_cor(aes(label=..rr.label..),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

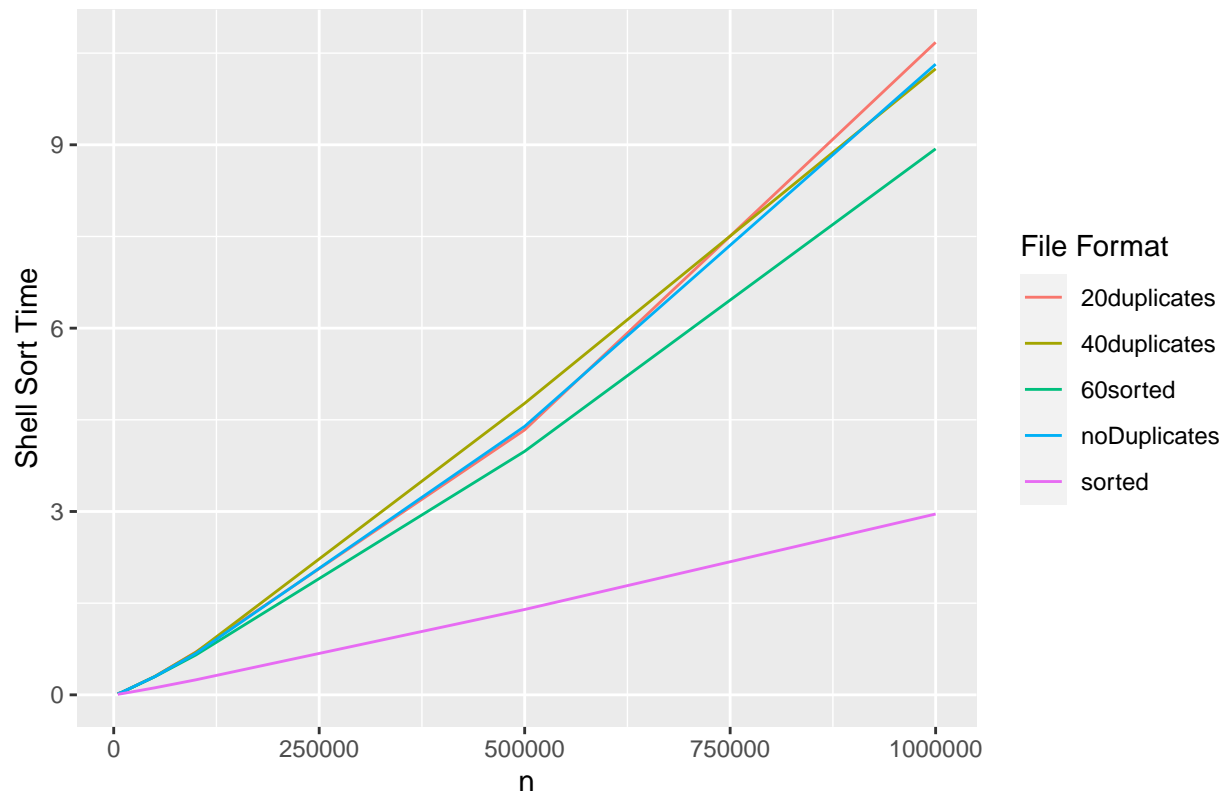


```
shellInts = subset(shellTimes, var_type == "int")
ggplot(shellInts, aes(x = size, y = shell_time, color = format)) +
  geom_line() +
  labs(title = "Shell Sort Time With Integer Data By Data Set Size and File Format", x = "n", y = "Shell")
  guides(color = guide_legend(title = "File Format"))
```



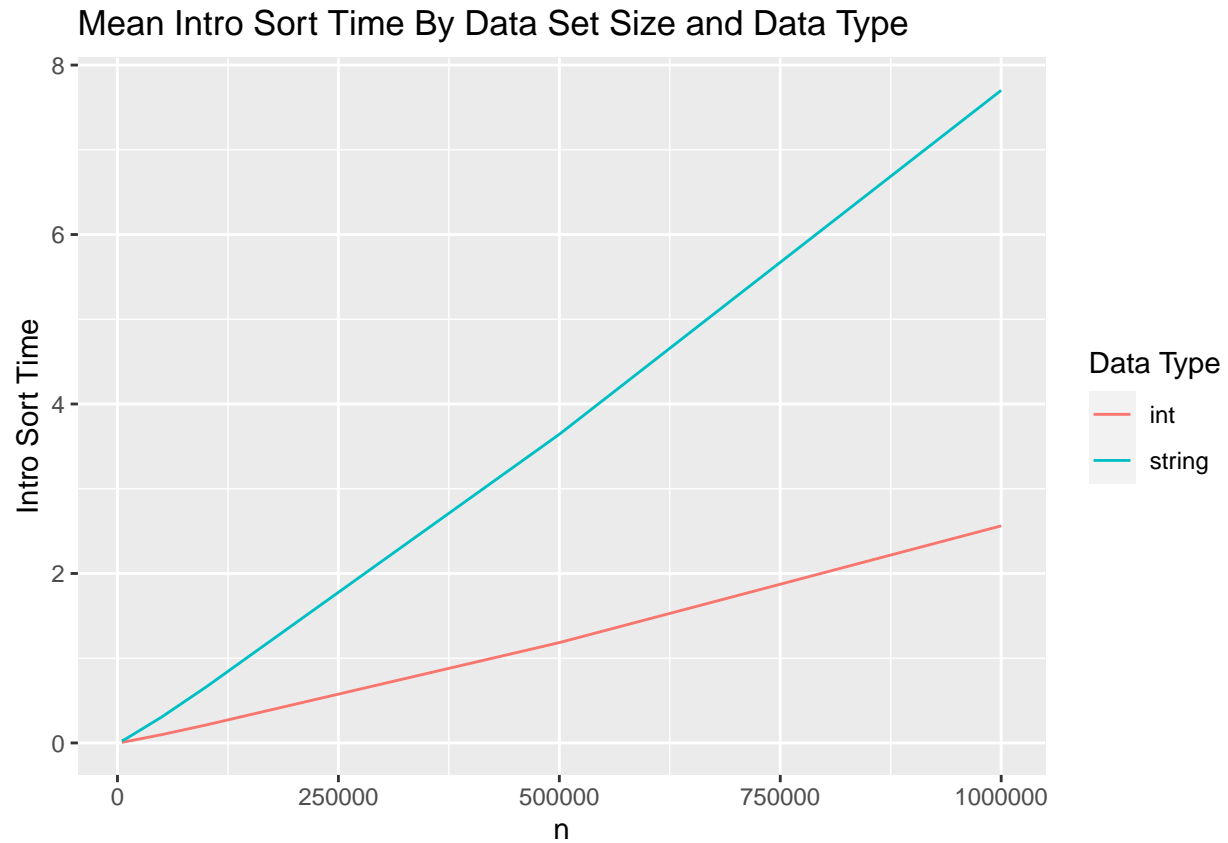
```
shellStrings = subset(shellTimes, var_type == "string")
ggplot(shellStrings, aes(x = size, y = shell_time, color = format)) +
  geom_line() +
  labs(title = "Shell Sort Time With String Data By Data Set Size and File Format", x = "n", y = "Shell")
guides(color = guide_legend(title = "File Format"))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

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



Intro Sort

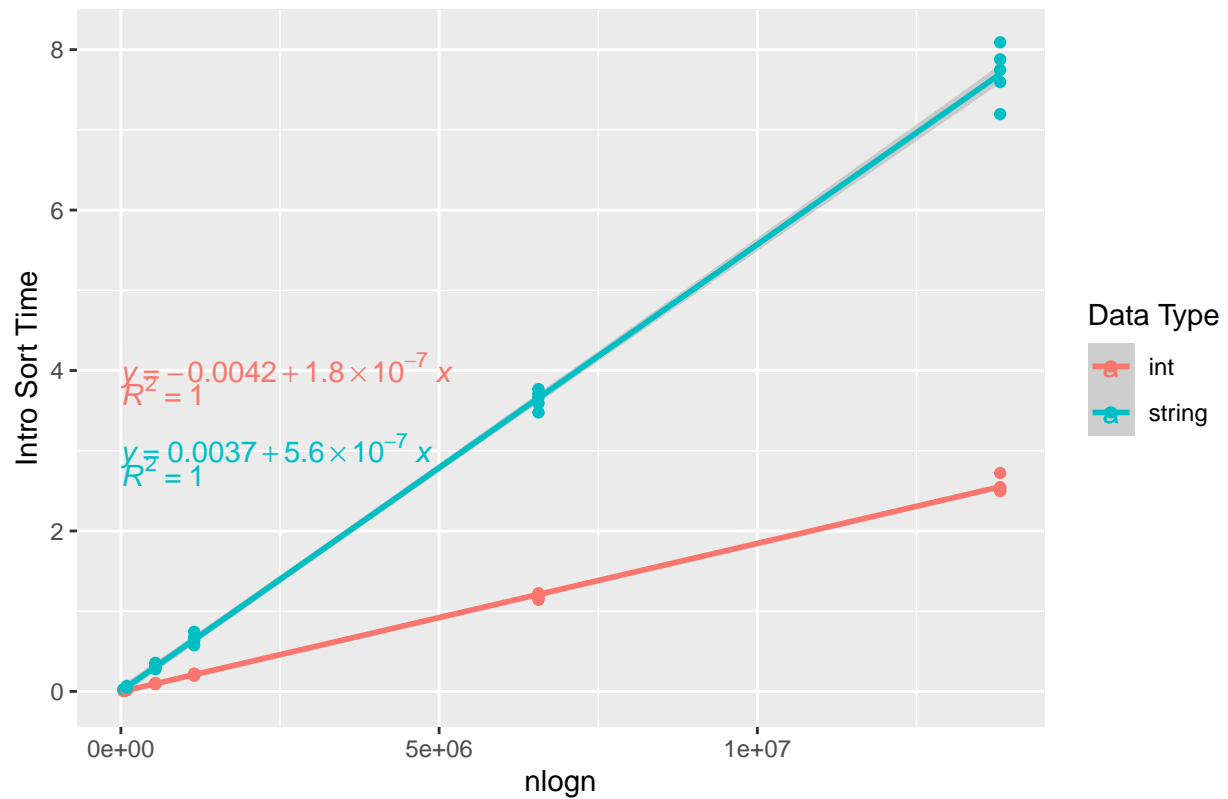
```
introTimes = aggregate(intro_time ~ var_type + size + nlogn + format, data = data, FUN = mean)
introTimes2 = aggregate(intro_time ~ var_type + size + nlogn, data = data, FUN = mean)
ggplot(introTimes2, aes(x = size, y = intro_time, color = var_type)) +
  geom_line() +
  labs(title = "Mean Intro Sort 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"))
```

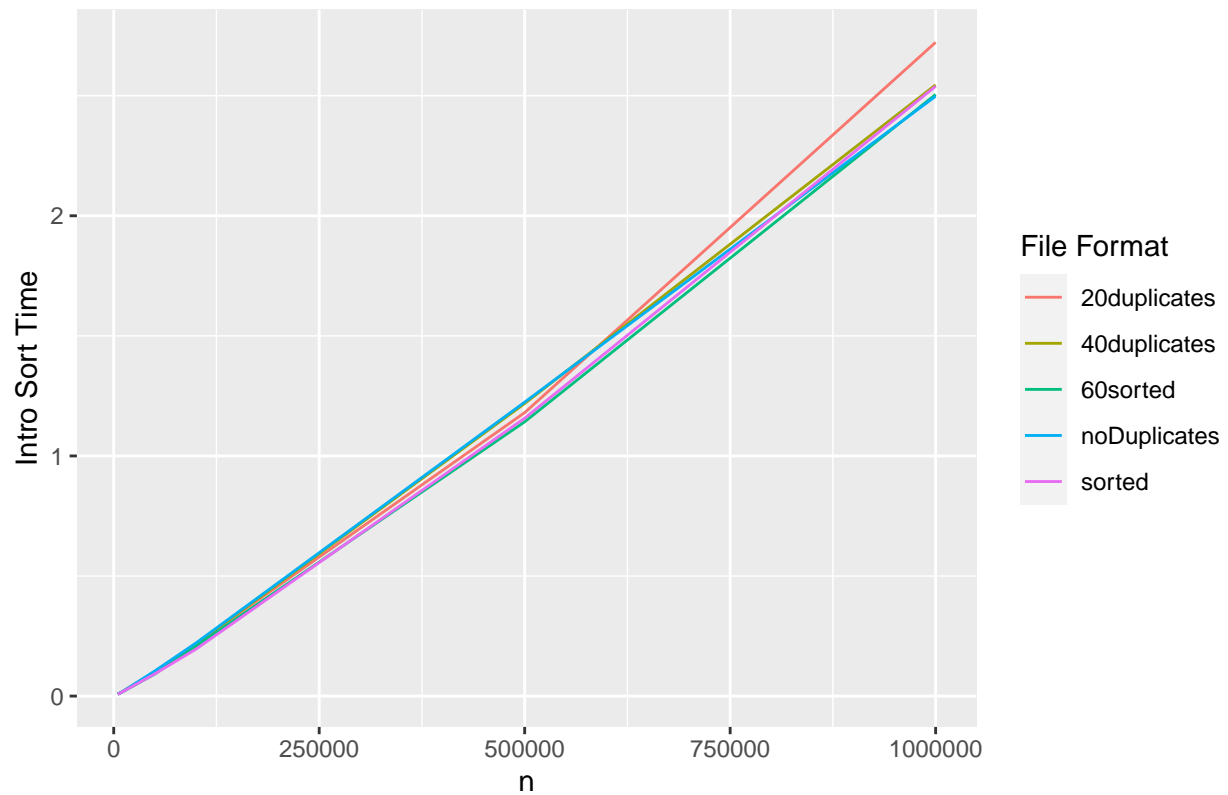
```
## 'geom_smooth()' using formula 'y ~ x'
```


Intro Sort Regression Models By Data Type

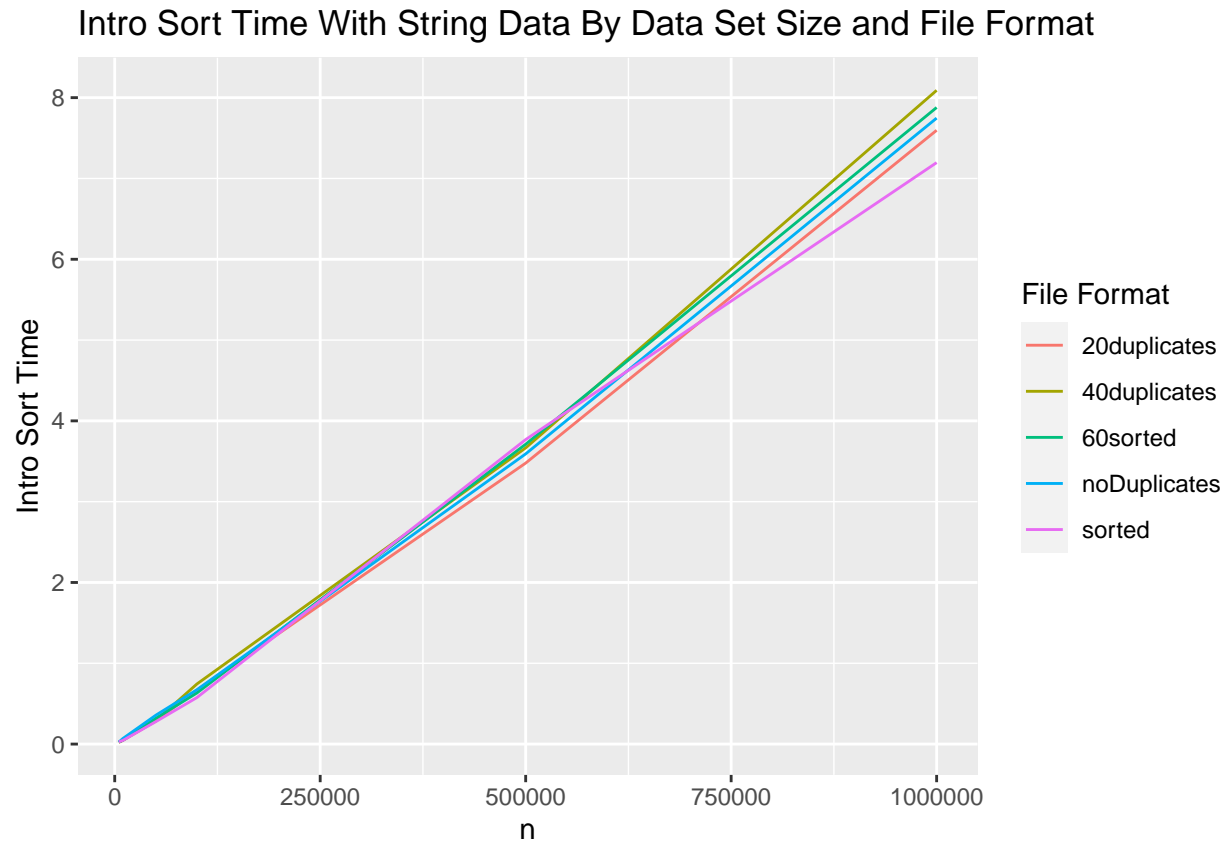


```
introInts = subset(introTimes, var_type == "int")
ggplot(introInts, aes(x = size, y = intro_time, color = format)) +
  geom_line() +
  labs(title = "Intro Sort Time With Integer Data By Data Set Size and File Format", x = "n", y = "Intro Sort Time")
guides(color = guide_legend(title = "File Format"))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

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



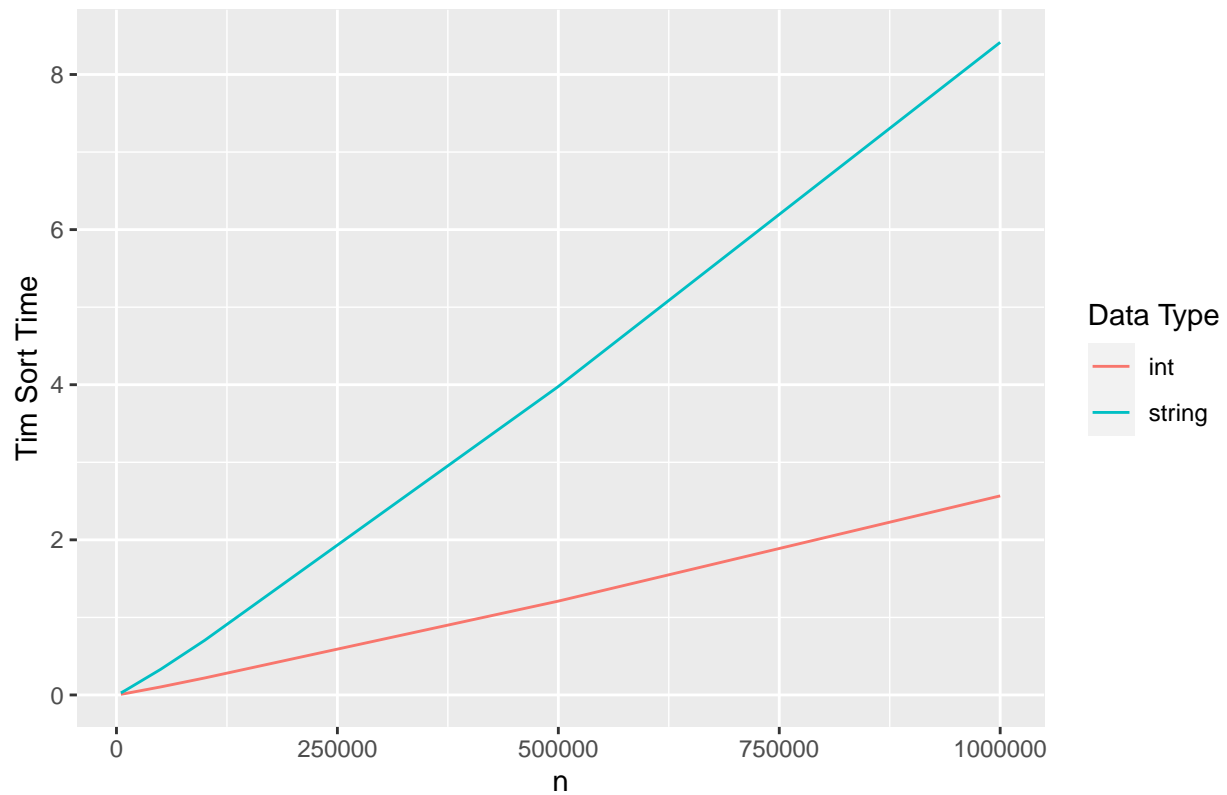
```
introStrings = subset(introTimes, var_type == "string")
ggplot(introStrings, aes(x = size, y = intro_time, color = format)) +
  geom_line() +
  labs(title = "Intro Sort Time With String Data By Data Set Size and File Format", x = "n", y = "Intro")
guides(color = guide_legend(title = "File Format"))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



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"))
```

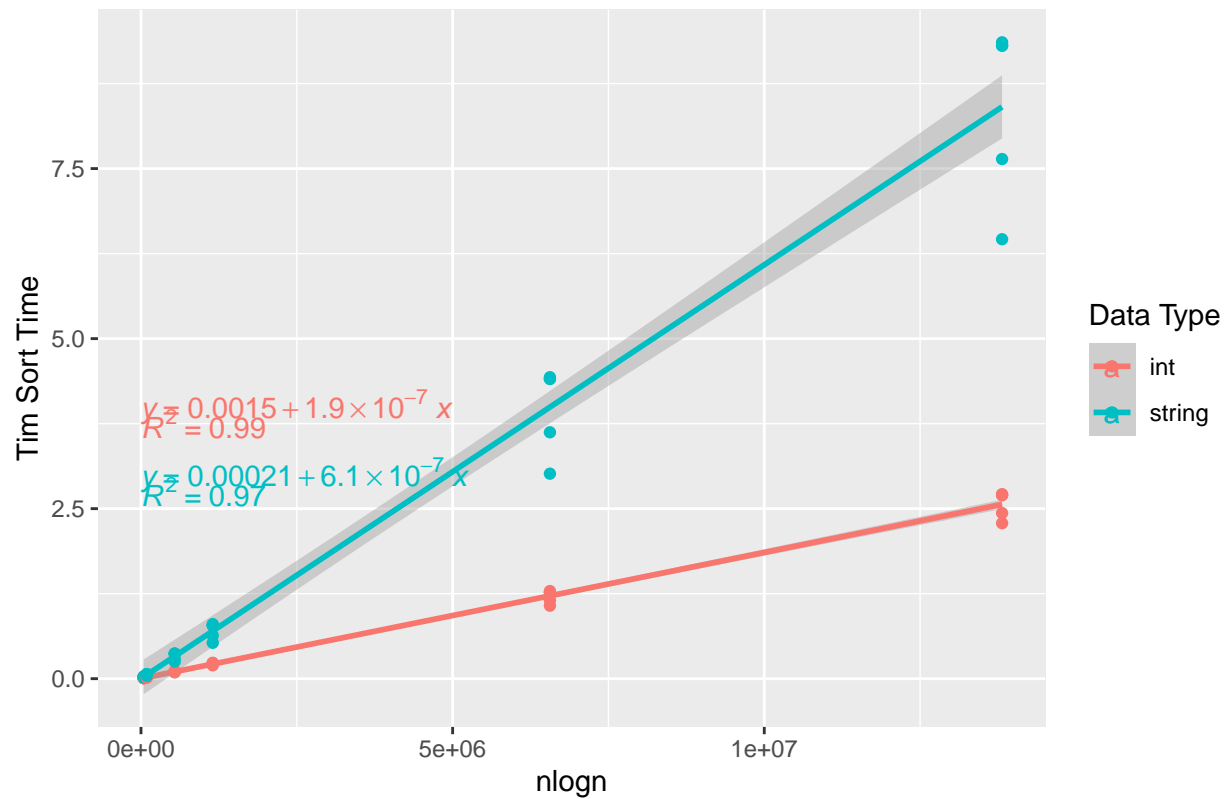
Mean Tim Sort Time By Data Set Size and Data Type



```
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"))
```

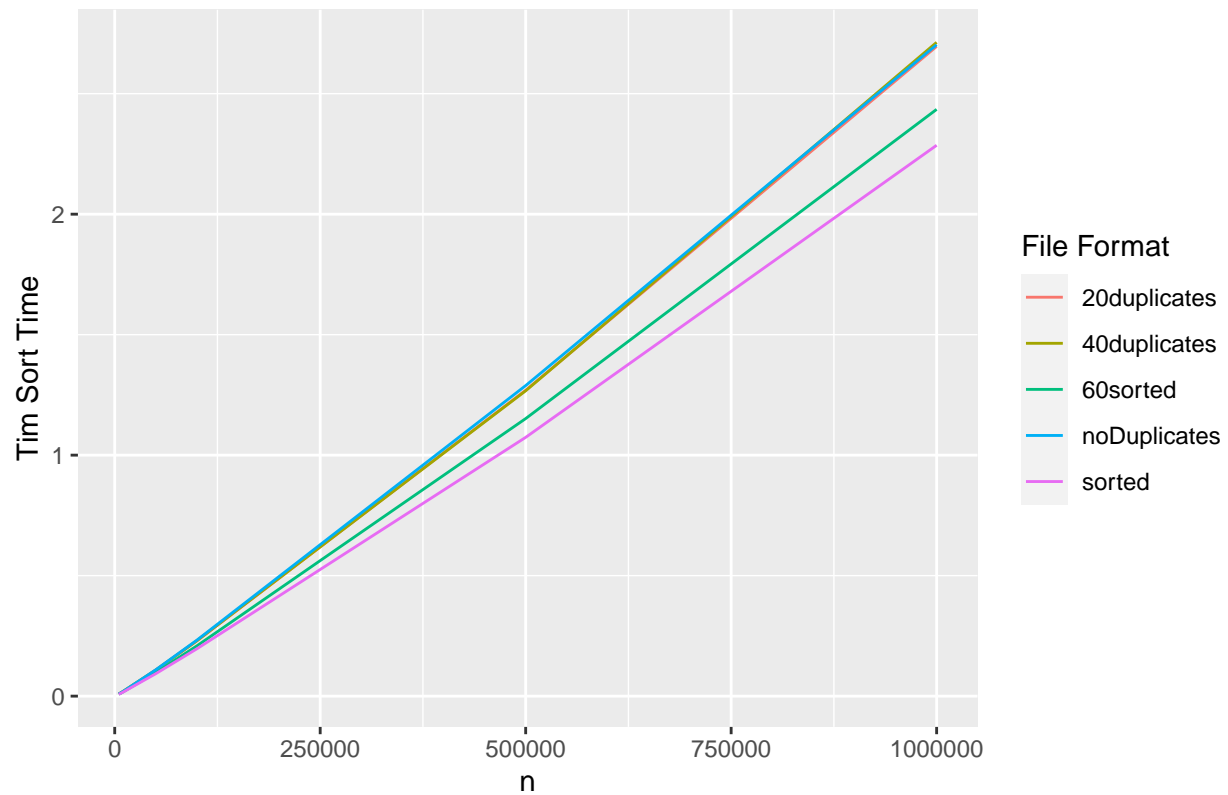
```
## 'geom_smooth()' using formula 'y ~ x'
```

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

