

Data Analytics 101 Course Project: Car Price Predictor

Wisdom Agbeve & Daniel Enesi

2024-11-27

Brief introduction of the project: This project aims to build predictive models for car prices using two datasets: Cars93 and CarPrice_Assignment

Load both cars data

```
cars93 <- read.csv("Cars93.csv")  
cars_assign = read.csv("CarPrice_Assignment.csv")
```

Standard deviations for cars__assign

```
sd(cars_assign$enginesize)
```

```
## [1] 41.64269
```

```
sd(cars_assign$horsepower)
```

```
## [1] 39.54417
```

```
sd(cars_assign$curbweight)
```

```
## [1] 520.6802
```

```
sd(cars_assign$price)
```

```
## [1] 7988.852
```

Standard deviations for cars93

```
sd(cars93$EngineSize)
```

```
## [1] 1.037363
```

```
sd(cars93$Horsepower)
```

```
## [1] 52.37441
```

```
sd(cars93$Weight)
```

```
## [1] 589.8965
```

```
knitr::opts_chunk$set(echo = TRUE)
library(ggplot2) # Ensure ggplot2 is loaded
```

Dataset 1 Summary

```
summary(cars_assign)
```

```
##      car_ID      symboling      CarName      fueltype
## Min.   : 1      Min.   :-2.0000      Length:205      Length:205
## 1st Qu.: 52      1st Qu.: 0.0000      Class :character  Class :character
## Median :103      Median : 1.0000      Mode  :character  Mode  :character
## Mean   :103      Mean   : 0.8341
## 3rd Qu.:154      3rd Qu.: 2.0000
## Max.   :205      Max.   : 3.0000
##      aspiration      doornumber      carbody      drivewheel
## Length:205      Length:205      Length:205      Length:205
## Class :character  Class :character  Class :character  Class :character
## Mode  :character  Mode  :character  Mode  :character  Mode  :character
##
##
##      enginelocation      wheelbase      carlength      carwidth
## Length:205      Min.   : 86.60      Min.   :141.1      Min.   :60.30
## Class :character  1st Qu.: 94.50      1st Qu.:166.3      1st Qu.:64.10
## Mode  :character  Median : 97.00      Median :173.2      Median :65.50
##                      Mean   : 98.76      Mean   :174.0      Mean   :65.91
##                      3rd Qu.:102.40      3rd Qu.:183.1      3rd Qu.:66.90
##                      Max.   :120.90      Max.   :208.1      Max.   :72.30
##      carheight      curbweight      enginetype      cylindernumber
## Min.   :47.80      Min.   :1488      Length:205      Length:205
## 1st Qu.:52.00      1st Qu.:2145      Class :character  Class :character
## Median :54.10      Median :2414      Mode  :character  Mode  :character
## Mean   :53.72      Mean   :2556
## 3rd Qu.:55.50      3rd Qu.:2935
## Max.   :59.80      Max.   :4066
##      enginesize      fuelsystem      boreratio      stroke
## Min.   : 61.0      Length:205      Min.   :2.54      Min.   :2.070
## 1st Qu.: 97.0      Class :character  1st Qu.:3.15      1st Qu.:3.110
## Median :120.0      Mode  :character  Median :3.31      Median :3.290
## Mean   :126.9                      Mean   :3.33      Mean   :3.255
## 3rd Qu.:141.0                      3rd Qu.:3.58      3rd Qu.:3.410
## Max.   :326.0                      Max.   :3.94      Max.   :4.170
##      compressionratio      horsepower      peakrpm      citympg
## Min.   : 7.00      Min.   : 48.0      Min.   :4150      Min.   :13.00
## 1st Qu.: 8.60      1st Qu.: 70.0      1st Qu.:4800      1st Qu.:19.00
## Median : 9.00      Median : 95.0      Median :5200      Median :24.00
## Mean   :10.14      Mean   :104.1      Mean   :5125      Mean   :25.22
## 3rd Qu.: 9.40      3rd Qu.:116.0      3rd Qu.:5500      3rd Qu.:30.00
```

```
## Max. :23.00 Max. :288.0 Max. :6600 Max. :49.00
## highwaympg price
## Min. :16.00 Min. : 5118
## 1st Qu.:25.00 1st Qu.: 7788
## Median :30.00 Median :10295
## Mean :30.75 Mean :13277
## 3rd Qu.:34.00 3rd Qu.:16503
## Max. :54.00 Max. :45400
```

Dataset 2 Summary

```
summary(cars93)
```

```
##      id      Manufacturer      Model      Type
## Min.   : 1  Length:93      Length:93      Length:93
## 1st Qu.:24  Class :character  Class :character  Class :character
## Median :47  Mode  :character  Mode  :character  Mode  :character
## Mean   :47
## 3rd Qu.:70
## Max.   :93
##
##      Min.Price      Price      Max.Price      MPG.city      MPG.highway
## Min.   : 6.70  Min.   : 7.40  Min.   : 7.9  Min.   :15.00  Min.   :20.00
## 1st Qu.:10.80  1st Qu.:12.20  1st Qu.:14.7  1st Qu.:18.00  1st Qu.:26.00
## Median :14.70  Median :17.70  Median :19.6  Median :21.00  Median :28.00
## Mean   :17.13  Mean   :19.51  Mean   :21.9  Mean   :22.37  Mean   :29.09
## 3rd Qu.:20.30  3rd Qu.:23.30  3rd Qu.:25.3  3rd Qu.:25.00  3rd Qu.:31.00
## Max.   :45.40  Max.   :61.90  Max.   :80.0  Max.   :46.00  Max.   :50.00
##
##      AirBags      DriveTrain      Cylinders      EngineSize
## Length:93      Length:93      Length:93      Min.   :1.000
## Class :character  Class :character  Class :character  1st Qu.:1.800
## Mode  :character  Mode  :character  Mode  :character  Median :2.400
##                                     Mean   :2.668
##                                     3rd Qu.:3.300
##                                     Max.   :5.700
##
##      Horsepower      RPM      Rev.per.mile  Man.trans.avail
## Min.   : 55.0  Min.   :3800  Min.   :1320  Length:93
## 1st Qu.:103.0  1st Qu.:4800  1st Qu.:1985  Class :character
## Median :140.0  Median :5200  Median :2340  Mode  :character
## Mean   :143.8  Mean   :5281  Mean   :2332
## 3rd Qu.:170.0  3rd Qu.:5750  3rd Qu.:2565
## Max.   :300.0  Max.   :6500  Max.   :3755
##
##      Fuel.tank.capacity  Passengers      Length      Wheelbase
## Min.   : 9.20  Min.   :2.000  Min.   :141.0  Min.   : 90.0
## 1st Qu.:14.50  1st Qu.:4.000  1st Qu.:174.0  1st Qu.: 98.0
## Median :16.40  Median :5.000  Median :183.0  Median :103.0
## Mean   :16.66  Mean   :5.086  Mean   :183.2  Mean   :103.9
## 3rd Qu.:18.80  3rd Qu.:6.000  3rd Qu.:192.0  3rd Qu.:110.0
## Max.   :27.00  Max.   :8.000  Max.   :219.0  Max.   :119.0
##
```

```
##      Width      Turn.circle  Rear.seat.room  Luggage.room      Weight
##  Min.   :60.00   Min.   :32.00   Min.   :19.00   Min.    : 6.00   Min.   :1695
##  1st Qu.:67.00   1st Qu.:37.00   1st Qu.:26.00   1st Qu.:12.00   1st Qu.:2620
##  Median :69.00   Median :39.00   Median :27.50   Median :14.00   Median :3040
##  Mean   :69.38   Mean   :38.96   Mean   :27.83   Mean   :13.89   Mean   :3073
##  3rd Qu.:72.00   3rd Qu.:41.00   3rd Qu.:30.00   3rd Qu.:15.00   3rd Qu.:3525
##  Max.   :78.00   Max.   :45.00   Max.   :36.00   Max.   :22.00   Max.   :4105
##
##              NA's      :2      NA's      :11
##      Origin      Make
## Length:93      Length:93
## Class :character Class :character
## Mode  :character Mode  :character
##
##
##
##
```

Scatter

```
cars93$Passengers
```

```
## [1] 5 5 5 6 4 6 6 6 5 6 5 5 5 4 6 7 8 6 2 6 6 6 5 5 6 7 6 4 5 6 4 5 5 4 4 7 5 6
## [39] 4 4 4 4 4 5 5 4 5 5 5 4 6 6 4 5 5 7 2 5 5 4 5 5 5 5 5 7 5 5 5 7 6 4 4 5 4 5
## [77] 6 5 5 4 5 5 4 5 4 5 7 4 7 5 4 5 5
```

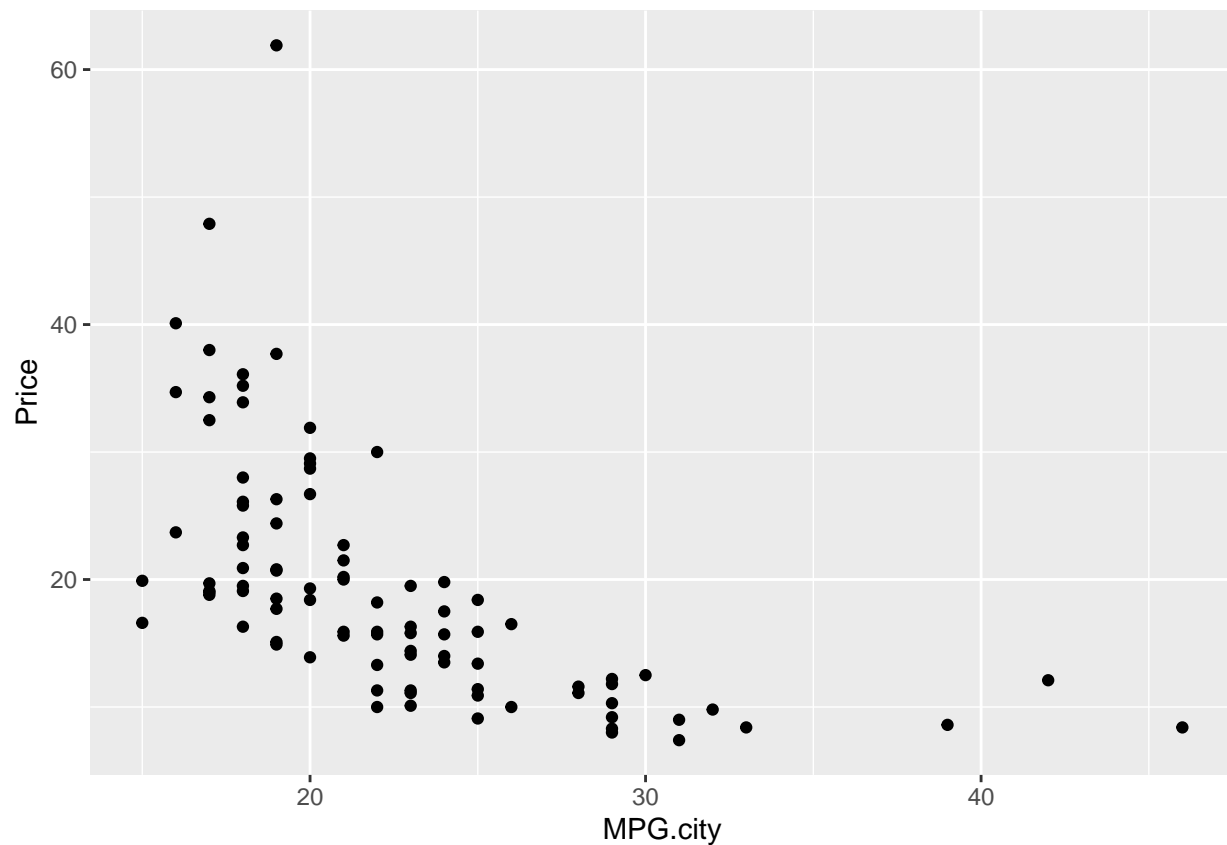
```
cars93$Cylinders
```

```
## [1] "4"      "6"      "6"      "6"      "4"      "4"      "6"      "6"
## [9] "6"      "8"      "8"      "4"      "4"      "6"      "4"      "6"
## [17] "6"      "8"      "8"      "6"      "4"      "6"      "4"      "4"
## [25] "4"      "6"      "4"      "6"      "4"      "6"      "4"      "4"
## [33] "4"      "4"      "4"      "6"      "6"      "8"      "3"      "4"
## [41] "4"      "4"      "4"      "4"      "4"      "4"      "4"      "8"
## [49] "6"      "6"      "6"      "8"      "4"      "4"      "4"      "6"
## [57] "rotary" "4"      "6"      "4"      "6"      "4"      "6"      "4"
## [65] "4"      "6"      "6"      "4"      "4"      "6"      "6"      "4"
## [73] "4"      "4"      "6"      "6"      "6"      "4"      "4"      "3"
## [81] "4"      "4"      "3"      "4"      "4"      "4"      "4"      "4"
## [89] "5"      "4"      "6"      "4"      "5"
```

```
columns93 <- c("MPG.city", "MPG.highway", "EngineSize", "Horsepower", "RPM", "Rev.per.mile", "Fuel.tank
length(columns93)
```

```
## [1] 13
```

```
ggplot(cars93, aes(x = MPG.city, y = Price)) +
  geom_point()
```



```
cars93[]
```

##	id	Manufacturer	Model	Type	Min.Price	Price	Max.Price	MPG.city
## 1	1	Acura	Integra	Small	12.9	15.9	18.8	25
## 2	2	Acura	Legend	Midsize	29.2	33.9	38.7	18
## 3	3	Audi	90	Compact	25.9	29.1	32.3	20
## 4	4	Audi	100	Midsize	30.8	37.7	44.6	19
## 5	5	BMW	535i	Midsize	23.7	30.0	36.2	22
## 6	6	Buick	Century	Midsize	14.2	15.7	17.3	22
## 7	7	Buick	LeSabre	Large	19.9	20.8	21.7	19
## 8	8	Buick	Roadmaster	Large	22.6	23.7	24.9	16
## 9	9	Buick	Riviera	Midsize	26.3	26.3	26.3	19
## 10	10	Cadillac	DeVille	Large	33.0	34.7	36.3	16
## 11	11	Cadillac	Seville	Midsize	37.5	40.1	42.7	16
## 12	12	Chevrolet	Cavalier	Compact	8.5	13.4	18.3	25
## 13	13	Chevrolet	Corsica	Compact	11.4	11.4	11.4	25
## 14	14	Chevrolet	Camaro	Sporty	13.4	15.1	16.8	19
## 15	15	Chevrolet	Lumina	Midsize	13.4	15.9	18.4	21
## 16	16	Chevrolet	Lumina_APV	Van	14.7	16.3	18.0	18
## 17	17	Chevrolet	Astro	Van	14.7	16.6	18.6	15
## 18	18	Chevrolet	Caprice	Large	18.0	18.8	19.6	17
## 19	19	Chevrolet	Corvette	Sporty	34.6	38.0	41.5	17
## 20	20	Chrysler	Concorde	Large	18.4	18.4	18.4	20
## 21	21	Chrysler	LeBaron	Compact	14.5	15.8	17.1	23
## 22	22	Chrysler	Imperial	Large	29.5	29.5	29.5	20
## 23	23	Dodge	Colt	Small	7.9	9.2	10.6	29

##	24	24	Dodge	Shadow	Small	8.4	11.3	14.2	23
##	25	25	Dodge	Spirit	Compact	11.9	13.3	14.7	22
##	26	26	Dodge	Caravan	Van	13.6	19.0	24.4	17
##	27	27	Dodge	Dynasty	Midsize	14.8	15.6	16.4	21
##	28	28	Dodge	Stealth	Sporty	18.5	25.8	33.1	18
##	29	29	Eagle	Summit	Small	7.9	12.2	16.5	29
##	30	30	Eagle	Vision	Large	17.5	19.3	21.2	20
##	31	31	Ford	Festiva	Small	6.9	7.4	7.9	31
##	32	32	Ford	Escort	Small	8.4	10.1	11.9	23
##	33	33	Ford	Tempo	Compact	10.4	11.3	12.2	22
##	34	34	Ford	Mustang	Sporty	10.8	15.9	21.0	22
##	35	35	Ford	Probe	Sporty	12.8	14.0	15.2	24
##	36	36	Ford	Aerostar	Van	14.5	19.9	25.3	15
##	37	37	Ford	Taurus	Midsize	15.6	20.2	24.8	21
##	38	38	Ford	Crown_Victoria	Large	20.1	20.9	21.7	18
##	39	39	Geo	Metro	Small	6.7	8.4	10.0	46
##	40	40	Geo	Storm	Sporty	11.5	12.5	13.5	30
##	41	41	Honda	Prelude	Sporty	17.0	19.8	22.7	24
##	42	42	Honda	Civic	Small	8.4	12.1	15.8	42
##	43	43	Honda	Accord	Compact	13.8	17.5	21.2	24
##	44	44	Hyundai	Excel	Small	6.8	8.0	9.2	29
##	45	45	Hyundai	Elantra	Small	9.0	10.0	11.0	22
##	46	46	Hyundai	Scoupe	Sporty	9.1	10.0	11.0	26
##	47	47	Hyundai	Sonata	Midsize	12.4	13.9	15.3	20
##	48	48	Infiniti	Q45	Midsize	45.4	47.9	50.4	17
##	49	49	Lexus	ES300	Midsize	27.5	28.0	28.4	18
##	50	50	Lexus	SC300	Midsize	34.7	35.2	35.6	18
##	51	51	Lincoln	Continental	Midsize	33.3	34.3	35.3	17
##	52	52	Lincoln	Town_Car	Large	34.4	36.1	37.8	18
##	53	53	Mazda	323	Small	7.4	8.3	9.1	29
##	54	54	Mazda	Protege	Small	10.9	11.6	12.3	28
##	55	55	Mazda	626	Compact	14.3	16.5	18.7	26
##	56	56	Mazda	MPV	Van	16.6	19.1	21.7	18
##	57	57	Mazda	RX-7	Sporty	32.5	32.5	32.5	17
##	58	58	Mercedes-Benz	190E	Compact	29.0	31.9	34.9	20
##	59	59	Mercedes-Benz	300E	Midsize	43.8	61.9	80.0	19
##	60	60	Mercury	Capri	Sporty	13.3	14.1	15.0	23
##	61	61	Mercury	Cougar	Midsize	14.9	14.9	14.9	19
##	62	62	Mitsubishi	Mirage	Small	7.7	10.3	12.9	29
##	63	63	Mitsubishi	Diamante	Midsize	22.4	26.1	29.9	18
##	64	64	Nissan	Sentra	Small	8.7	11.8	14.9	29
##	65	65	Nissan	Altima	Compact	13.0	15.7	18.3	24
##	66	66	Nissan	Quest	Van	16.7	19.1	21.5	17
##	67	67	Nissan	Maxima	Midsize	21.0	21.5	22.0	21
##	68	68	Oldsmobile	Achieva	Compact	13.0	13.5	14.0	24
##	69	69	Oldsmobile	Cutlass_Ciera	Midsize	14.2	16.3	18.4	23
##	70	70	Oldsmobile	Silhouette	Van	19.5	19.5	19.5	18
##	71	71	Oldsmobile	Eighty-Eight	Large	19.5	20.7	21.9	19
##	72	72	Plymouth	Laser	Sporty	11.4	14.4	17.4	23
##	73	73	Pontiac	LeMans	Small	8.2	9.0	9.9	31
##	74	74	Pontiac	Sunbird	Compact	9.4	11.1	12.8	23
##	75	75	Pontiac	Firebird	Sporty	14.0	17.7	21.4	19
##	76	76	Pontiac	Grand_Prix	Midsize	15.4	18.5	21.6	19
##	77	77	Pontiac	Bonneville	Large	19.4	24.4	29.4	19

## 78 78	Saab	900	Compact	20.3	28.7	37.1	20
## 79 79	Saturn	SL	Small	9.2	11.1	12.9	28
## 80 80	Subaru	Justy	Small	7.3	8.4	9.5	33
## 81 81	Subaru	Loyale	Small	10.5	10.9	11.3	25
## 82 82	Subaru	Legacy	Compact	16.3	19.5	22.7	23
## 83 83	Suzuki	Swift	Small	7.3	8.6	10.0	39
## 84 84	Toyota	Tercel	Small	7.8	9.8	11.8	32
## 85 85	Toyota	Celica	Sporty	14.2	18.4	22.6	25
## 86 86	Toyota	Camry	Midsize	15.2	18.2	21.2	22
## 87 87	Toyota	Previa	Van	18.9	22.7	26.6	18
## 88 88	Volkswagen	Fox	Small	8.7	9.1	9.5	25
## 89 89	Volkswagen	Eurovan	Van	16.6	19.7	22.7	17
## 90 90	Volkswagen	Passat	Compact	17.6	20.0	22.4	21
## 91 91	Volkswagen	Corrado	Sporty	22.9	23.3	23.7	18
## 92 92	Volvo	240	Compact	21.8	22.7	23.5	21
## 93 93	Volvo	850	Midsize	24.8	26.7	28.5	20
##	MPG.highway	AirBags	DriveTrain	Cylinders	EngineSize	Horsepower	
## 1	31	None	Front	4	1.8	140	
## 2	25	Driver & Passenger	Front	6	3.2	200	
## 3	26	Driver only	Front	6	2.8	172	
## 4	26		Front	6	2.8	172	
## 5	30	Driver only	Rear	4	3.5	208	
## 6	31	Driver only	Front	4	2.2	110	
## 7	28	Driver only	Front	6	3.8	170	
## 8	25	Driver only	Rear	6	5.7	180	
## 9	27	Driver only	Front	6	3.8	170	
## 10	25	Driver only	Front	8	4.9	200	
## 11	25	Driver & Passenger	Front	8	4.6	295	
## 12	36	None	Front	4	2.2	110	
## 13	34	Driver only	Front	4	2.2	110	
## 14	28	Driver & Passenger	Rear	6	3.4	160	
## 15	29	None	Front	4	2.2	110	
## 16	23	None	Front	6	3.8	170	
## 17	20	None	4WD	6	4.3	165	
## 18	26	Driver only	Rear	8	5.0	170	
## 19	25	Driver only	Rear	8	5.7	300	
## 20	28	Driver & Passenger	Front	6	3.3	153	
## 21	28	Driver & Passenger	Front	4	3.0	141	
## 22	26		Front	6	3.3	147	
## 23	33	None	Front	4	1.5	92	
## 24	29	Driver only	Front	4	2.2	93	
## 25	27	Driver only	Front	4	2.5	100	
## 26	21	Driver only	4WD	6	3.0	142	
## 27	27	Driver only	Front	4	2.5	100	
## 28	24	Driver only	4WD	6	3.0	300	
## 29	33	None	Front	4	1.5	92	
## 30	28	Driver & Passenger	Front	6	3.5	214	
## 31	33	None	Front	4	1.3	63	
## 32	30	None	Front	4	1.8	127	
## 33	27	None	Front	4	2.3	96	
## 34	29	Driver only	Rear	4	2.3	105	
## 35	30	Driver only	Front	4	2.0	115	
## 36	20	Driver only	4WD	6	3.0	145	
## 37	30	Driver only	Front	6	3.0	140	

## 38	26	Driver only	Rear	8	4.6	190
## 39	50	None	Front	3	1.0	55
## 40	36	Driver only	Front	4	1.6	90
## 41	31	Driver & Passenger	Front	4	2.3	160
## 42	46	Driver only	Front	4	1.5	102
## 43	31	Driver & Passenger	Front	4	2.2	140
## 44	33	None	Front	4	1.5	81
## 45	29	None	Front	4	1.8	124
## 46	34		Front	4	1.5	92
## 47	27	None	Front	4	2.0	128
## 48	22	Driver only	Rear	8	4.5	278
## 49	24	Driver only	Front	6	3.0	185
## 50	23	Driver & Passenger	Rear	6	3.0	225
## 51	26	Driver & Passenger	Front	6	3.8	160
## 52	26		Rear	8	4.6	210
## 53	37	None	Front	4	1.6	82
## 54	36	None	Front	4	1.8	103
## 55	34	Driver only	Front	4	2.5	164
## 56	24	None	4WD	6	3.0	155
## 57	25	Driver only	Rear	rotary	1.3	255
## 58	29	Driver only	Rear	4	2.3	130
## 59	25	Driver & Passenger	Rear	6	3.2	217
## 60	26	Driver only	Front	4	1.6	100
## 61	26	None	Rear	6	3.8	140
## 62	33	None	Front	4	1.5	92
## 63	24	Driver only	Front	6	3.0	202
## 64	33	Driver only	Front	4	1.6	110
## 65	30	Driver only	Front	4	2.4	150
## 66	23	None	Front	6	3.0	151
## 67	26	Driver only	Front	6	3.0	160
## 68	31	None	Front	4	2.3	155
## 69	31	Driver only	Front	4	2.2	110
## 70	23	None	Front	6	3.8	170
## 71	28	Driver only	Front	6	3.8	170
## 72	30	None	4WD	4	1.8	92
## 73	41	None	Front	4	1.6	74
## 74	31	None	Front	4	2.0	110
## 75	28	Driver & Passenger	Rear	6	3.4	160
## 76	27	None	Front	6	3.4	200
## 77	28	Driver & Passenger	Front	6	3.8	170
## 78	26	Driver only	Front	4	2.1	140
## 79	38	Driver only	Front	4	1.9	85
## 80	37	None	4WD	3	1.2	73
## 81	30	None	4WD	4	1.8	90
## 82	30	Driver only	4WD	4	2.2	130
## 83	43	None	Front	3	1.3	70
## 84	37	Driver only	Front	4	1.5	82
## 85	32	Driver only	Front	4	2.2	135
## 86	29	Driver only	Front	4	2.2	130
## 87	22	Driver only	4WD	4	2.4	138
## 88	33	None	Front	4	1.8	81
## 89	21	None	Front	5	2.5	109
## 90	30	None	Front	4	2.0	134
## 91	25	None	Front	6	2.8	178

## 92	28	Driver only	Rear	4	2.3	114
## 93	28	Driver & Passenger	Front	5	2.4	168
##	RPM	Rev.per.mile	Man.trans.avail	Fuel.tank.capacity	Passengers	Length
## 1	6300	2890	Yes	13.2	5	177
## 2	5500	2335	Yes	18.0	5	195
## 3	5500	2280	Yes	16.9	5	180
## 4	5500	2535	Yes	21.1	6	193
## 5	5700	2545	Yes	21.1	4	186
## 6	5200	2565	No	16.4	6	189
## 7	4800	1570	No	18.0	6	200
## 8	4000	1320	No	23.0	6	216
## 9	4800	1690	No	18.8	5	198
## 10	4100	1510	No	18.0	6	206
## 11	6000	1985	No	20.0	5	204
## 12	5200	2380	Yes	15.2	5	182
## 13	5200	2665	Yes	15.6	5	184
## 14	4600	1805	Yes	15.5	4	193
## 15	5200	2595	No	16.5	6	198
## 16	4800	1690	No	20.0	7	178
## 17	4000	1790	No	27.0	8	194
## 18	4200	1350	No	23.0	6	214
## 19	5000	1450	Yes	20.0	2	179
## 20	5300	1990	No	18.0	6	203
## 21	5000	2090	No	16.0	6	183
## 22	4800	1785	No	16.0	6	203
## 23	6000	3285	Yes	13.2	5	174
## 24	4800	2595	Yes	14.0	5	172
## 25	4800	2535	Yes	16.0	6	181
## 26	5000	1970	No	20.0	7	175
## 27	4800	2465	No	16.0	6	192
## 28	6000	2120	Yes	19.8	4	180
## 29	6000	2505	Yes	13.2	5	174
## 30	5800	1980	No	18.0	6	202
## 31	5000	3150	Yes	10.0	4	141
## 32	6500	2410	Yes	13.2	5	171
## 33	4200	2805	Yes	15.9	5	177
## 34	4600	2285	Yes	15.4	4	180
## 35	5500	2340	Yes	15.5	4	179
## 36	4800	2080	Yes	21.0	7	176
## 37	4800	1885	No	16.0	5	192
## 38	4200	1415	No	20.0	6	212
## 39	5700	3755	Yes	10.6	4	151
## 40	5400	3250	Yes	12.4	4	164
## 41	5800	2855	Yes	15.9	4	175
## 42	5900	2650	Yes	11.9	4	173
## 43	5600	2610	Yes	17.0	4	185
## 44	5500	2710	Yes	11.9	5	168
## 45	6000	2745	Yes	13.7	5	172
## 46	5550	2540	Yes	11.9	4	166
## 47	6000	2335	Yes	17.2	5	184
## 48	6000	1955	No	22.5	5	200
## 49	5200	2325	Yes	18.5	5	188
## 50	6000	2510	Yes	20.6	4	191
## 51	4400	1835	No	18.4	6	205

## 52	4600	1840	No	20.0	6	219	
## 53	5000	2370	Yes	13.2	4	164	
## 54	5500	2220	Yes	14.5	5	172	
## 55	5600	2505	Yes	15.5	5	184	
## 56	5000	2240	No	19.6	7	190	
## 57	6500	2325	Yes	20.0	2	169	
## 58	5100	2425	Yes	14.5	5	175	
## 59	5500	2220	No	18.5	5	187	
## 60	5750	2475	Yes	11.1	4	166	
## 61	3800	1730	No	18.0	5	199	
## 62	6000	2505	Yes	13.2	5	172	
## 63	6000	2210	No	19.0	5	190	
## 64	6000	2435	Yes	13.2	5	170	
## 65	5600	2130	Yes	15.9	5	181	
## 66	4800	2065	No	20.0	7	190	
## 67	5200	2045	No	18.5	5	188	
## 68	6000	2380	No	15.2	5	188	
## 69	5200	2565	No	16.5	5	190	
## 70	4800	1690	No	20.0	7	194	
## 71	4800	1570	No	18.0	6	201	
## 72	5000	2360	Yes	15.9	4	173	
## 73	5600	3130	Yes	13.2	4	177	
## 74	5200	2665	Yes	15.2	5	181	
## 75	4600	1805	Yes	15.5	4	196	
## 76	5000	1890	Yes	16.5	5	195	
## 77	4800	1565	No	18.0	6	177	
## 78	6000	2910	Yes	18.0	5	184	
## 79	5000	2145	Yes	12.8	5	176	
## 80	5600	2875	Yes	9.2	4	146	
## 81	5200	3375	Yes	15.9	5	175	
## 82	5600	2330	Yes	15.9	5	179	
## 83	6000	3360	Yes	10.6	4	161	
## 84	5200	3505	Yes	11.9	5	162	
## 85	5400	2405	Yes	15.9	4	174	
## 86	5400	2340	Yes	18.5	5	188	
## 87	5000	2515	Yes	19.8	7	187	
## 88	5500	2550	Yes	12.4	4	163	
## 89	4500	2915	Yes	21.1	7	187	
## 90	5800	2685	Yes	18.5	5	180	
## 91	5800	2385	Yes	18.5	4	159	
## 92	5400	2215	Yes	15.8	5	190	
## 93	6200	2310	Yes	19.3	5	184	
##	Wheelbase	Width	Turn.circle	Rear.seat.room	Luggage.room	Weight	Origin
## 1	102	68	37	26.5	11	2705	non-USA
## 2	115	71	38	30.0	15	3560	non-USA
## 3	102	67	37	28.0	14	3375	non-USA
## 4	106	70	37	31.0	17	3405	non-USA
## 5	109	69	39	27.0	13	3640	non-USA
## 6	105	69	41	28.0	16	2880	USA
## 7	111	74	42	30.5	17	3470	USA
## 8	116	78	45	30.5	21	4105	USA
## 9	108	73	41	26.5	14	3495	USA
## 10	114	73	43	35.0	18	3620	USA
## 11	111	74	44	31.0	14	3935	USA

## 12	101	66	38	25.0	13	2490	USA
## 13	103	68	39	26.0	14	2785	USA
## 14	101	74	43	25.0	13	3240	USA
## 15	108	71	40	28.5	16	3195	USA
## 16	110	74	44	30.5	NA	3715	USA
## 17	111	78	42	33.5	NA	4025	USA
## 18	116	77	42	29.5	20	3910	USA
## 19	96	74	43	NA	NA	3380	USA
## 20	113	74	40	31.0	15	3515	USA
## 21	104	68	41	30.5	14	3085	USA
## 22	110	69	44	36.0	17	3570	USA
## 23	98	66	32	26.5	11	2270	USA
## 24	97	67	38	26.5	13	2670	USA
## 25	104	68	39	30.5	14	2970	USA
## 26	112	72	42	26.5	NA	3705	USA
## 27	105	69	42	30.5	16	3080	USA
## 28	97	72	40	20.0	11	3805	USA
## 29	98	66	36	26.5	11	2295	USA
## 30	113	74	40	30.0	15	3490	USA
## 31	90	63	33	26.0	12	1845	USA
## 32	98	67	36	28.0	12	2530	USA
## 33	100	68	39	27.5	13	2690	USA
## 34	101	68	40	24.0	12	2850	USA
## 35	103	70	38	23.0	18	2710	USA
## 36	119	72	45	30.0	NA	3735	USA
## 37	106	71	40	27.5	18	3325	USA
## 38	114	78	43	30.0	21	3950	USA
## 39	93	63	34	27.5	10	1695	non-USA
## 40	97	67	37	24.5	11	2475	non-USA
## 41	100	70	39	23.5	8	2865	non-USA
## 42	103	67	36	28.0	12	2350	non-USA
## 43	107	67	41	28.0	14	3040	non-USA
## 44	94	63	35	26.0	11	2345	non-USA
## 45	98	66	36	28.0	12	2620	non-USA
## 46	94	64	34	23.5	9	2285	non-USA
## 47	104	69	41	31.0	14	2885	non-USA
## 48	113	72	42	29.0	15	4000	non-USA
## 49	103	70	40	27.5	14	3510	non-USA
## 50	106	71	39	25.0	9	3515	non-USA
## 51	109	73	42	30.0	19	3695	USA
## 52	117	77	45	31.5	22	4055	USA
## 53	97	66	34	27.0	16	2325	non-USA
## 54	98	66	36	26.5	13	2440	non-USA
## 55	103	69	40	29.5	14	2970	non-USA
## 56	110	72	39	27.5	NA	3735	non-USA
## 57	96	69	37	NA	NA	2895	non-USA
## 58	105	67	34	26.0	12	2920	non-USA
## 59	110	69	37	27.0	15	3525	non-USA
## 60	95	65	36	19.0	6	2450	USA
## 61	113	73	38	28.0	15	3610	USA
## 62	98	67	36	26.0	11	2295	non-USA
## 63	107	70	43	27.5	14	3730	non-USA
## 64	96	66	33	26.0	12	2545	non-USA
## 65	103	67	40	28.5	14	3050	non-USA

## 66	112	74	41	27.0	NA	4100	non-USA
## 67	104	69	41	28.5	14	3200	non-USA
## 68	103	67	39	28.0	14	2910	USA
## 69	105	70	42	28.0	16	2890	USA
## 70	110	74	44	30.5	NA	3715	USA
## 71	111	74	42	31.5	17	3470	USA
## 72	97	67	39	24.5	8	2640	USA
## 73	99	66	35	25.5	17	2350	USA
## 74	101	66	39	25.0	13	2575	USA
## 75	101	75	43	25.0	13	3240	USA
## 76	108	72	41	28.5	16	3450	USA
## 77	111	74	43	30.5	18	3495	USA
## 78	99	67	37	26.5	14	2775	non-USA
## 79	102	68	40	26.5	12	2495	USA
## 80	90	60	32	23.5	10	2045	non-USA
## 81	97	65	35	27.5	15	2490	non-USA
## 82	102	67	37	27.0	14	3085	non-USA
## 83	93	63	34	27.5	10	1965	non-USA
## 84	94	65	36	24.0	11	2055	non-USA
## 85	99	69	39	23.0	13	2950	non-USA
## 86	103	70	38	28.5	15	3030	non-USA
## 87	113	71	41	35.0	NA	3785	non-USA
## 88	93	63	34	26.0	10	2240	non-USA
## 89	115	72	38	34.0	NA	3960	non-USA
## 90	103	67	35	31.5	14	2985	non-USA
## 91	97	66	36	26.0	15	2810	non-USA
## 92	104	67	37	29.5	14	2985	non-USA
## 93	105	69	38	30.0	15	3245	non-USA
##							
##			Make				
## 1			Acura Integra				
## 2			Acura Legend				
## 3			Audi 90				
## 4			Audi 100				
## 5			BMW 535i				
## 6			Buick Century				
## 7			Buick LeSabre				
## 8			Buick Roadmaster				
## 9			Buick Riviera				
## 10			Cadillac DeVille				
## 11			Cadillac Seville				
## 12			Chevrolet Cavalier				
## 13			Chevrolet Corsica				
## 14			Chevrolet Camaro				
## 15			Chevrolet Lumina				
## 16			Chevrolet Lumina_APV				
## 17			Chevrolet Astro				
## 18			Chevrolet Caprice				
## 19			Chevrolet Corvette				
## 20			Chrysler Concorde				
## 21			Chrysler LeBaron				
## 22			Chrysler Imperial				
## 23			Dodge Colt				
## 24			Dodge Shadow				
## 25			Dodge Spirit				

## 26	Dodge Caravan
## 27	Dodge Dynasty
## 28	Dodge Stealth
## 29	Eagle Summit
## 30	Eagle Vision
## 31	Ford Festiva
## 32	Ford Escort
## 33	Ford Tempo
## 34	Ford Mustang
## 35	Ford Probe
## 36	Ford Aerostar
## 37	Ford Taurus
## 38	Ford Crown_Victoria
## 39	Geo Metro
## 40	Geo Storm
## 41	Honda Prelude
## 42	Honda Civic
## 43	Honda Accord
## 44	Hyundai Excel
## 45	Hyundai Elantra
## 46	Hyundai Scoupe
## 47	Hyundai Sonata
## 48	Infiniti Q45
## 49	Lexus ES300
## 50	Lexus SC300
## 51	Lincoln Continental
## 52	Lincoln Town_Car
## 53	Mazda 323
## 54	Mazda Protege
## 55	Mazda 626
## 56	Mazda MPV
## 57	Mazda RX-7
## 58	Mercedes-Benz 190E
## 59	Mercedes-Benz 300E
## 60	Mercury Capri
## 61	Mercury Cougar
## 62	Mitsubishi Mirage
## 63	Mitsubishi Diamante
## 64	Nissan Sentra
## 65	Nissan Altima
## 66	Nissan Quest
## 67	Nissan Maxima
## 68	Oldsmobile Achieva
## 69	Oldsmobile Cutlass_Ciera
## 70	Oldsmobile Silhouette
## 71	Oldsmobile Eighty-Eight
## 72	Plymouth Laser
## 73	Pontiac LeMans
## 74	Pontiac Sunbird
## 75	Pontiac Firebird
## 76	Pontiac Grand_Prix
## 77	Pontiac Bonneville
## 78	Saab 900
## 79	Saturn SL

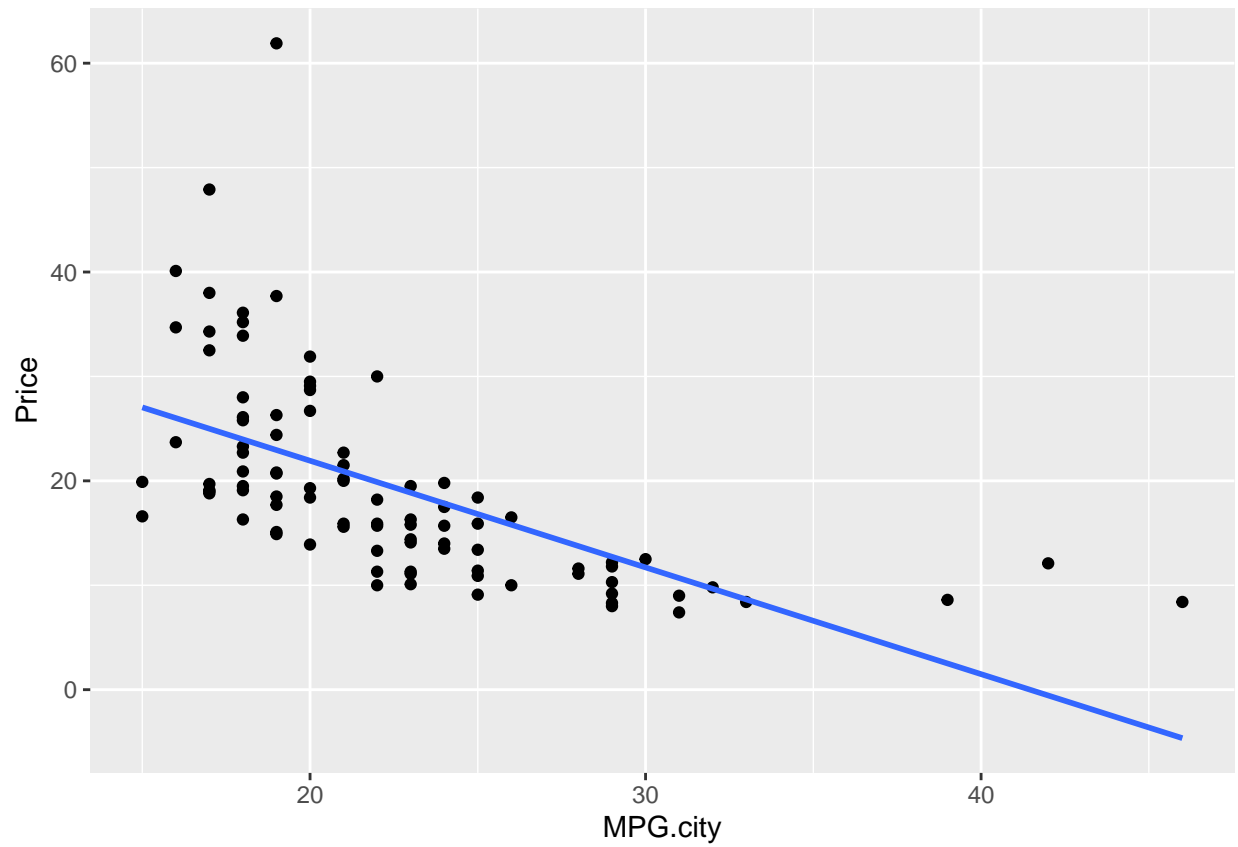
```
## 80          Subaru Justy
## 81          Subaru Loyale
## 82          Subaru Legacy
## 83          Suzuki Swift
## 84          Toyota Tercel
## 85          Toyota Celica
## 86          Toyota Camry
## 87          Toyota Previa
## 88          Volkswagen Fox
## 89          Volkswagen Eurovan
## 90          Volkswagen Passat
## 91          Volkswagen Corrado
## 92          Volvo 240
## 93          Volvo 850
```

```
for (var in columns93)
{
  print(var);
  p <- ggplot(cars93, aes_string(x = var, y = "Price")) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE);
  print(p);
  correlation <- cor(cars93$Price, cars93[[var]]);
  print(correlation)
}
```

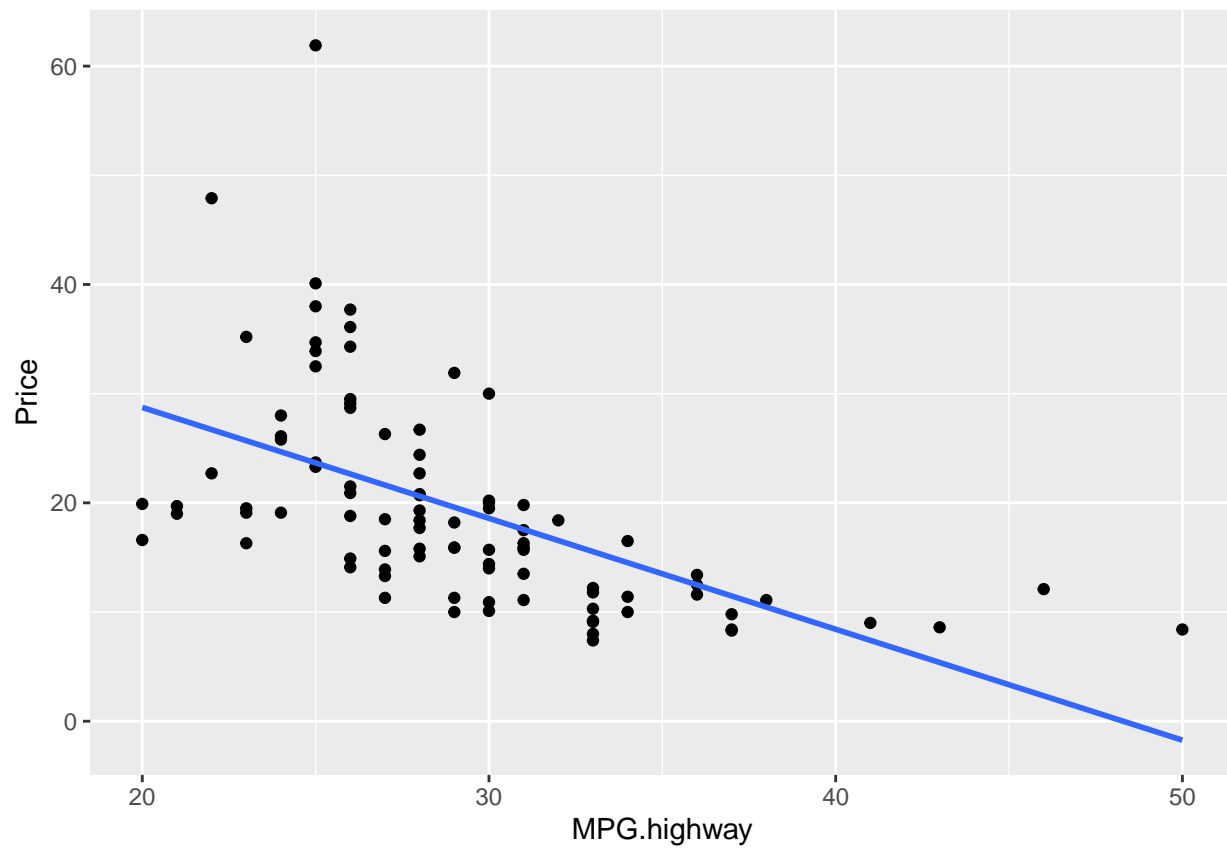
```
## [1] "MPG.city"
```

```
## Warning: 'aes_string()' was deprecated in ggplot2 3.0.0.
## i Please use tidy evaluation idioms with 'aes()'.
## i See also 'vignette("ggplot2-in-packages")' for more information.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.
```

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

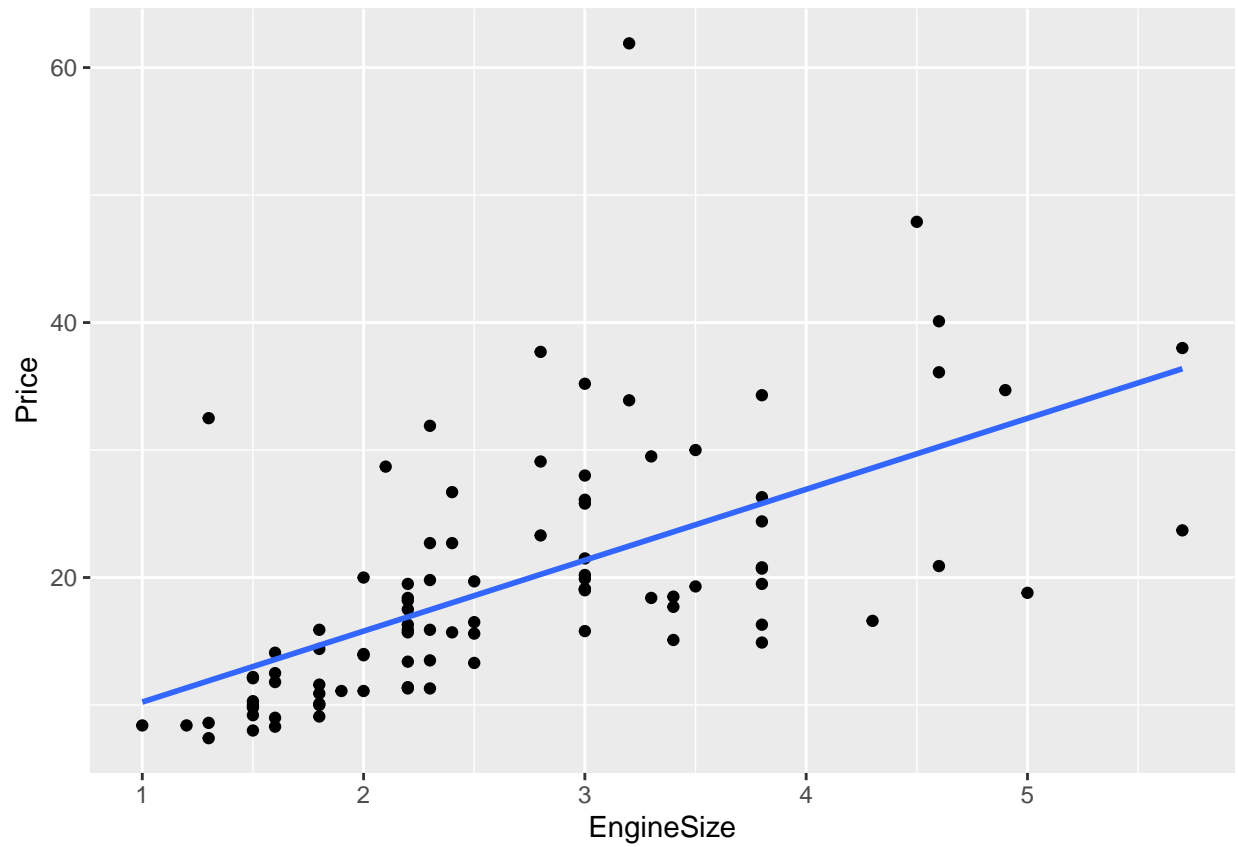


```
## [1] -0.5945622  
## [1] "MPG.highway"  
  
## 'geom_smooth()' using formula = 'y ~ x'
```



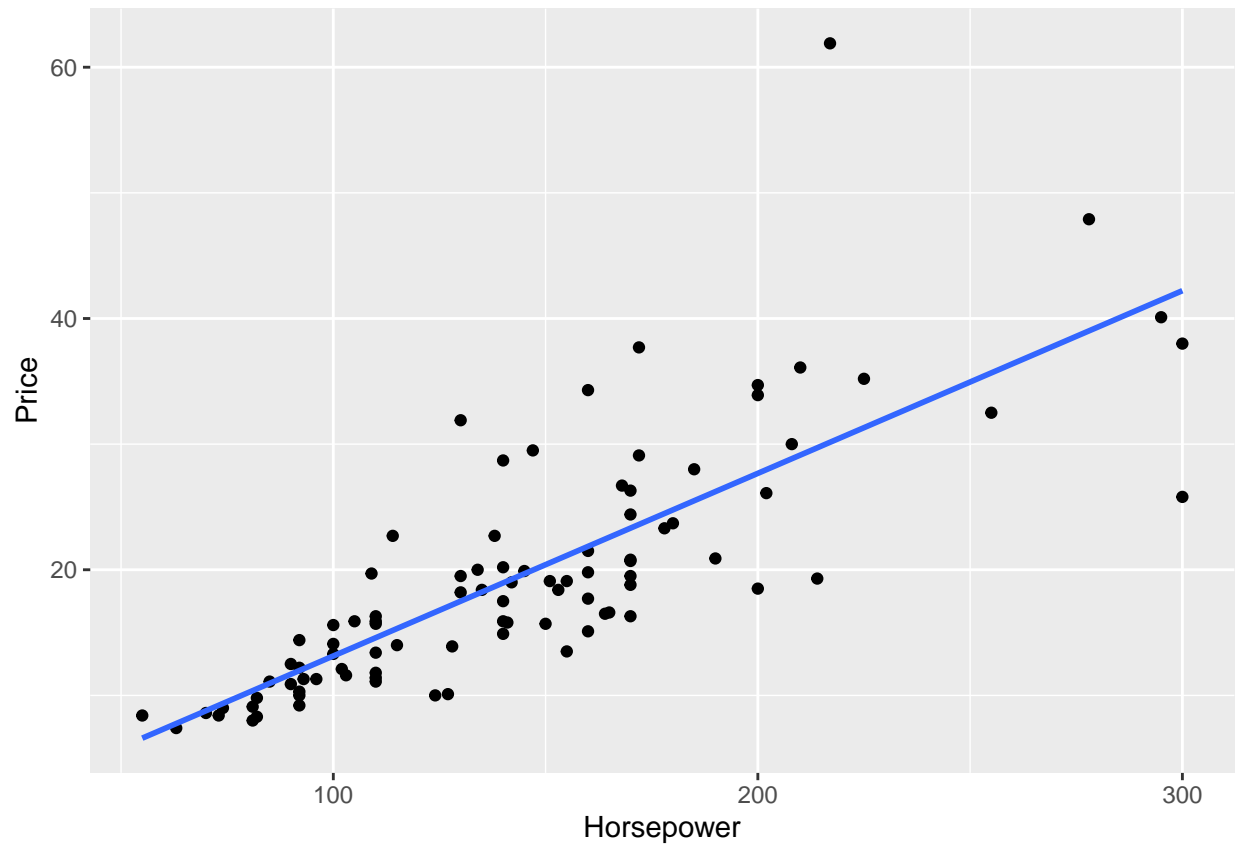
```
## [1] -0.5606804  
## [1] "EngineSize"
```

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

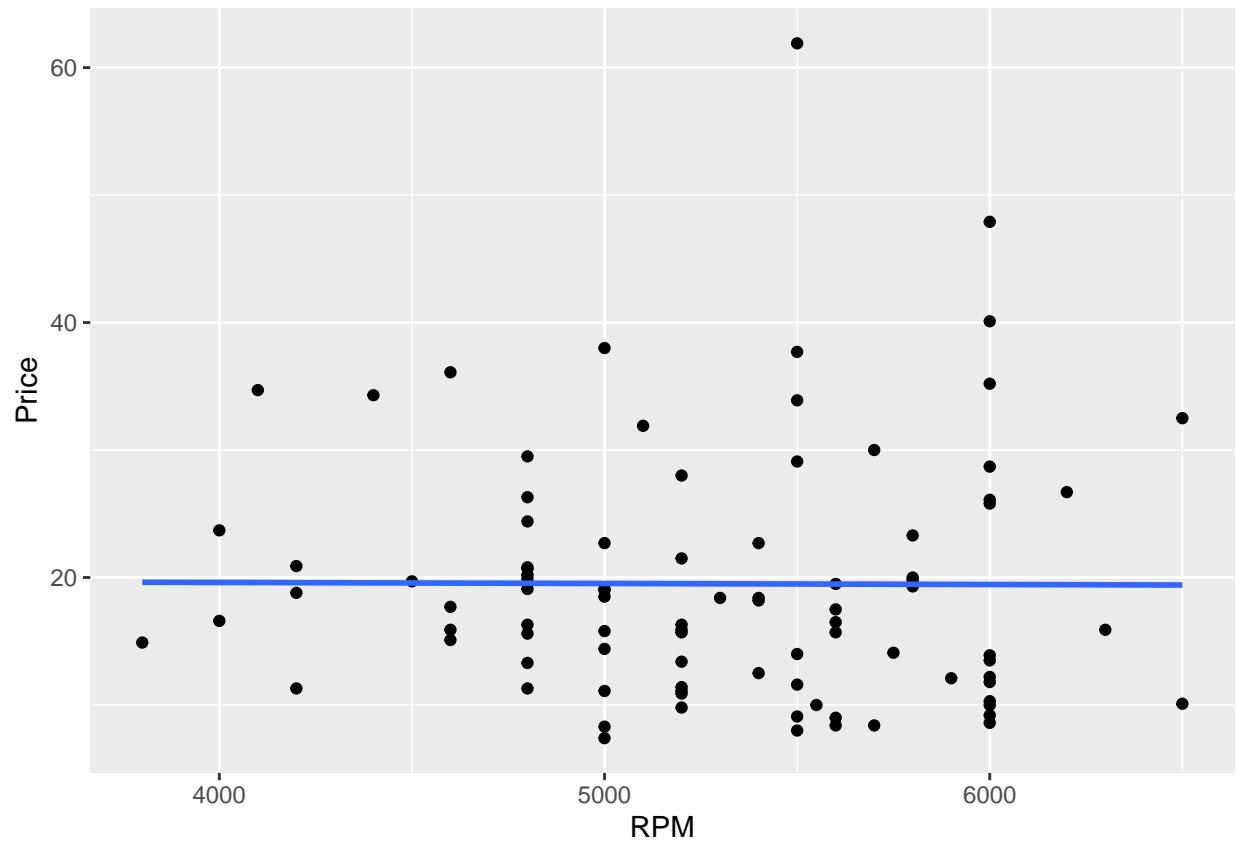
```
## [1] 0.5974254  
## [1] "Horsepower"
```

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



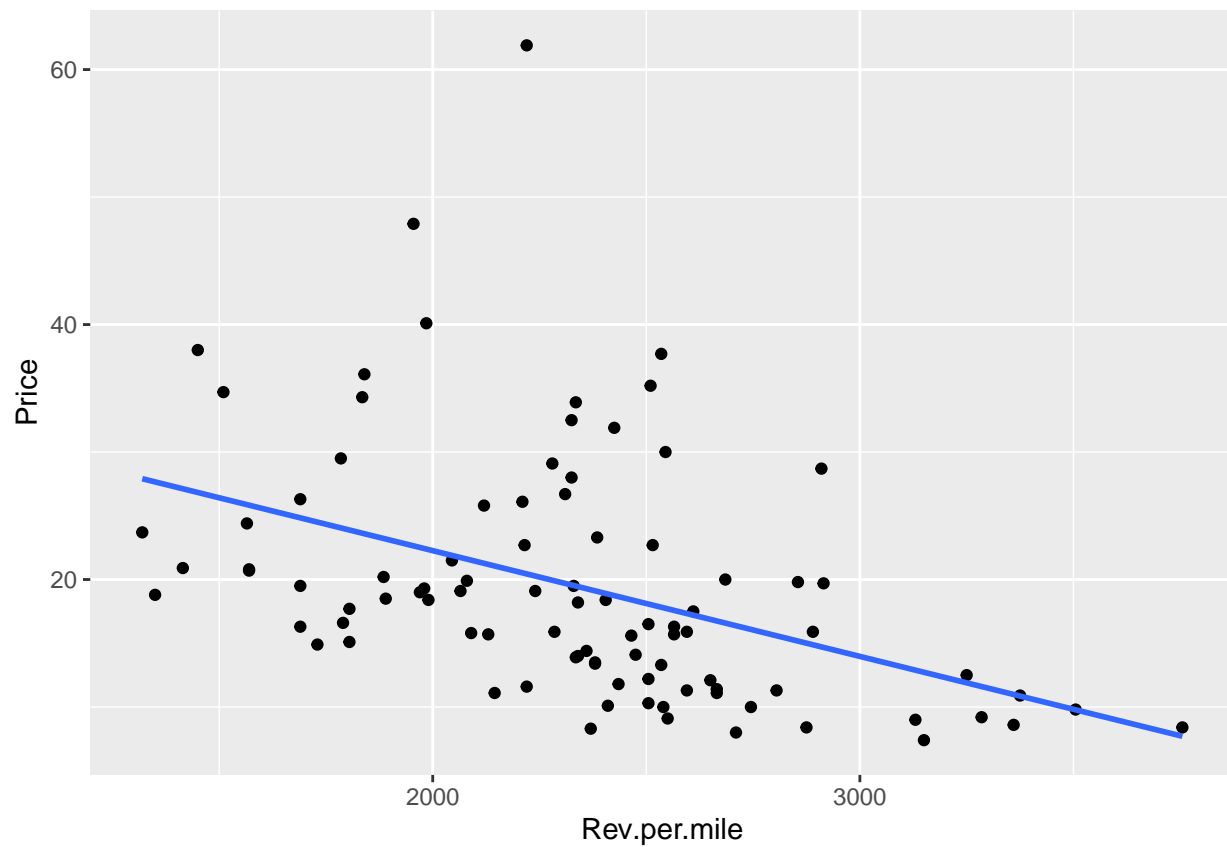
```
## [1] 0.7882176  
## [1] "RPM"
```

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



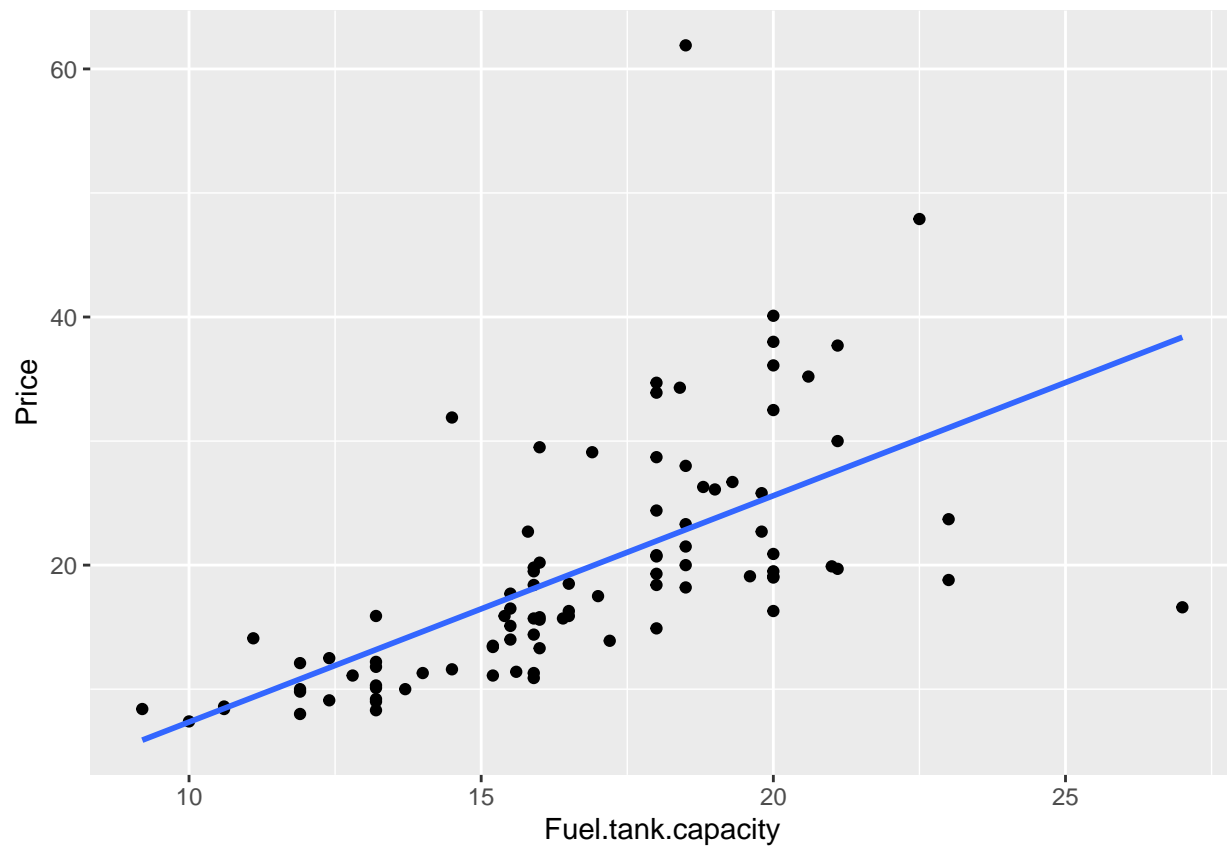
```
## [1] -0.004954931
## [1] "Rev.per.mile"

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



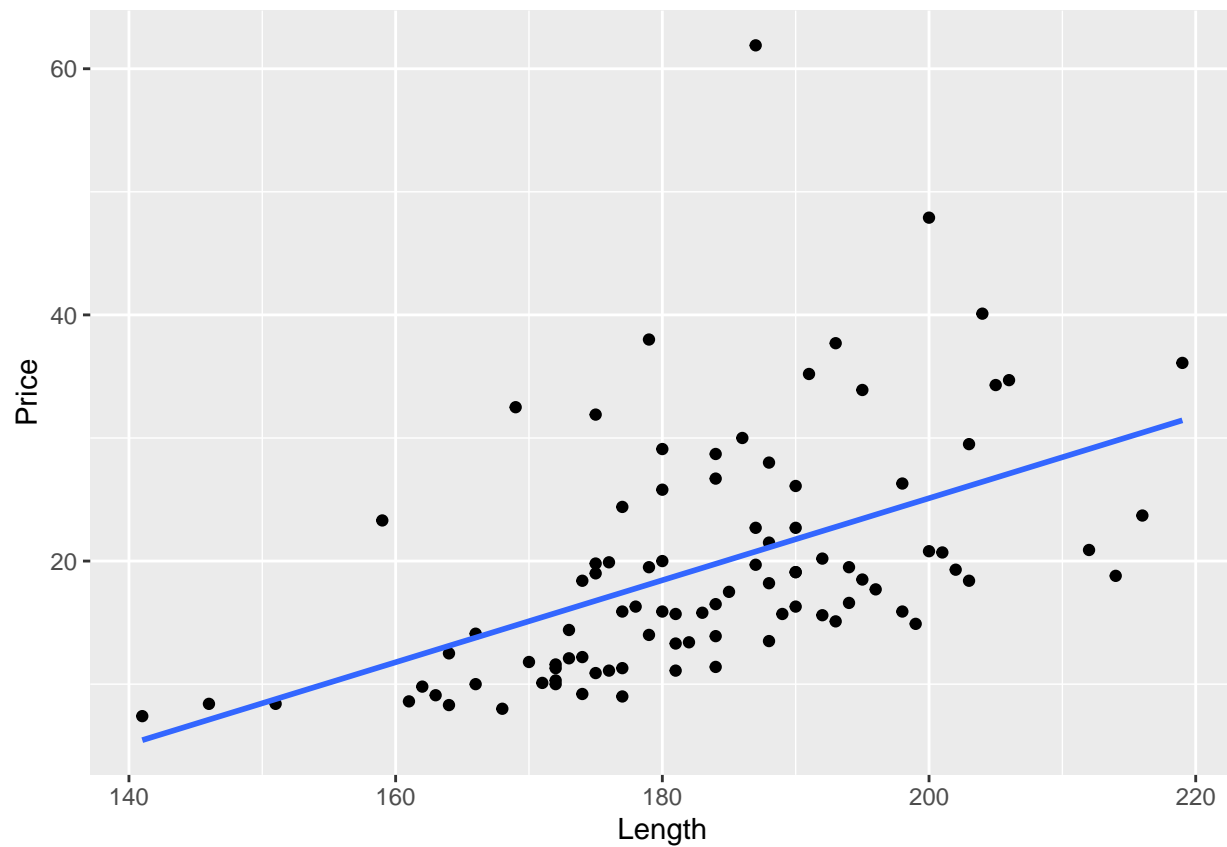
```
## [1] -0.4263951
## [1] "Fuel.tank.capacity"

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



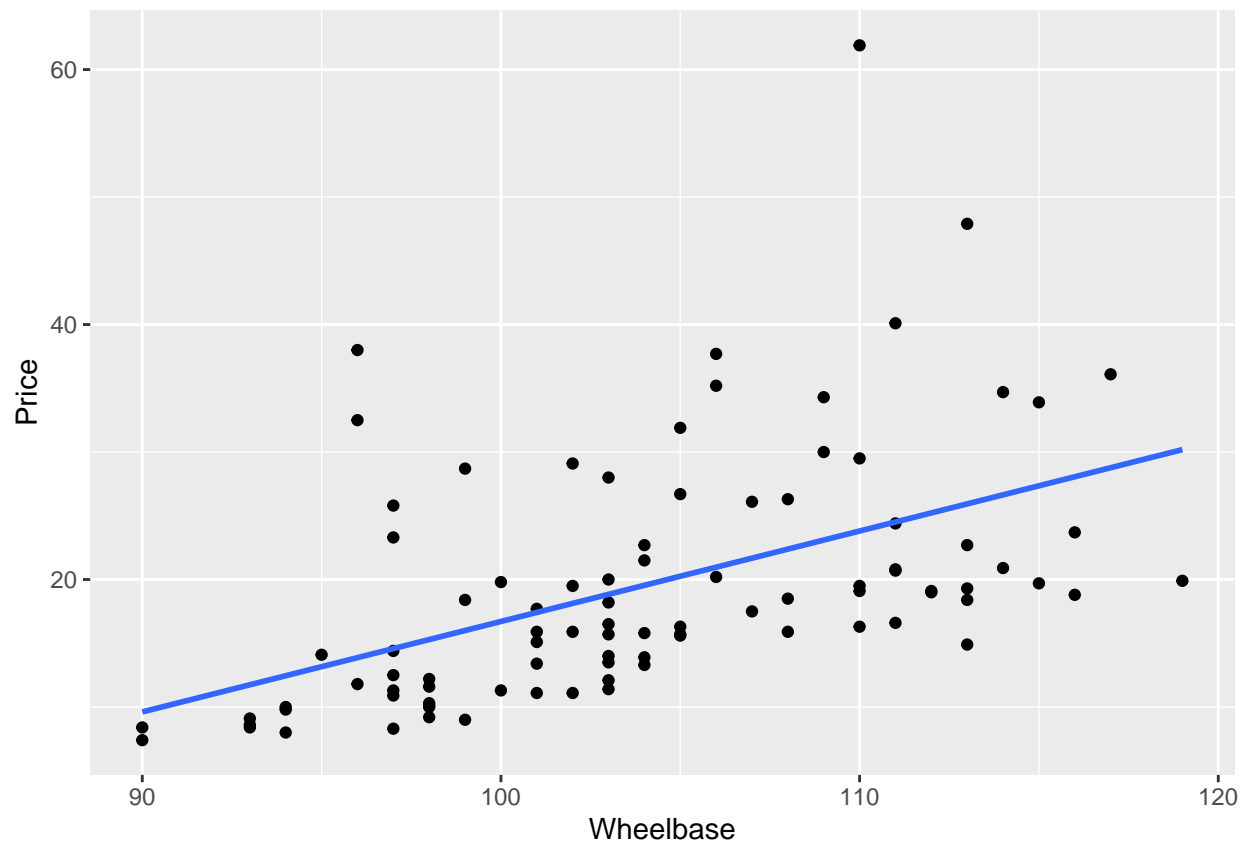
```
## [1] 0.61948  
## [1] "Length"
```

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



```
## [1] 0.5036284  
## [1] "Wheelbase"
```

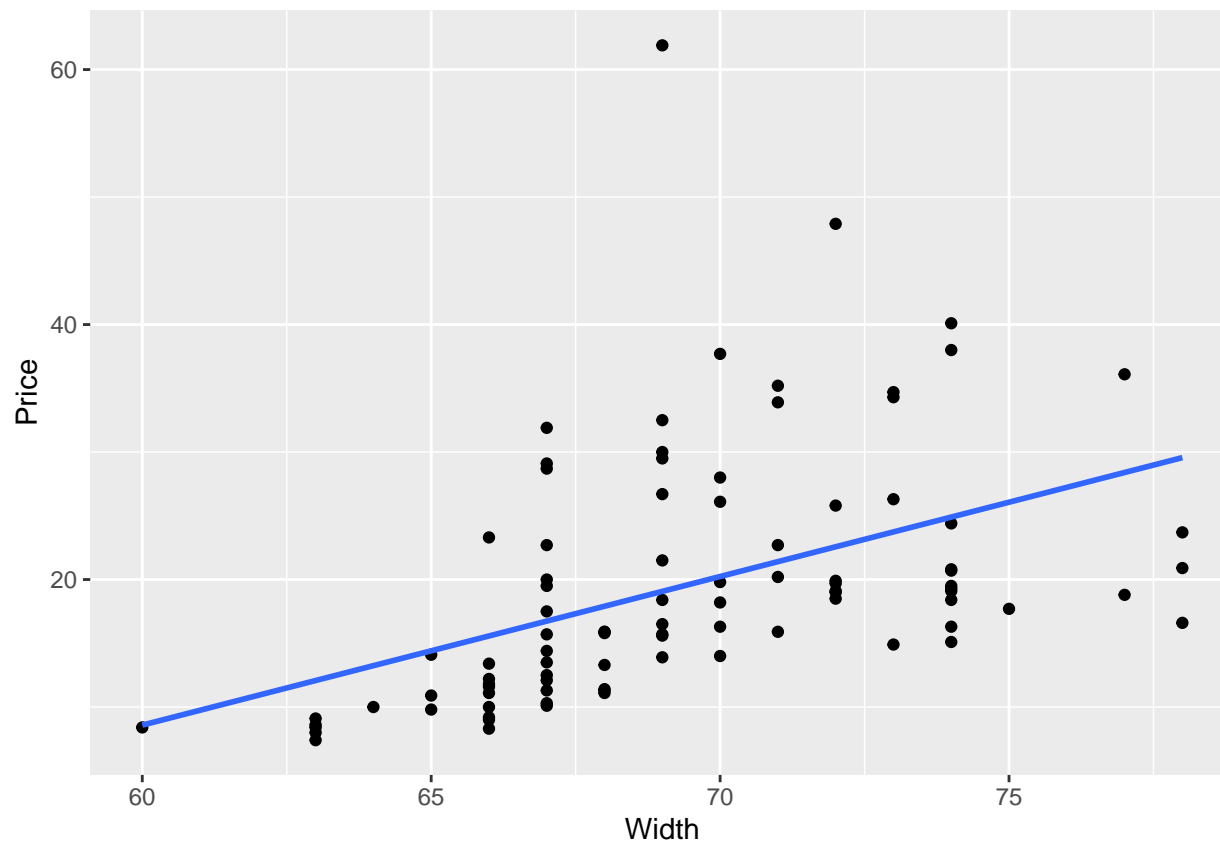
```
## 'geom_smooth()' using formula = 'y ~ x'
```



```
## [1] 0.5008642
```

```
## [1] "Width"
```

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

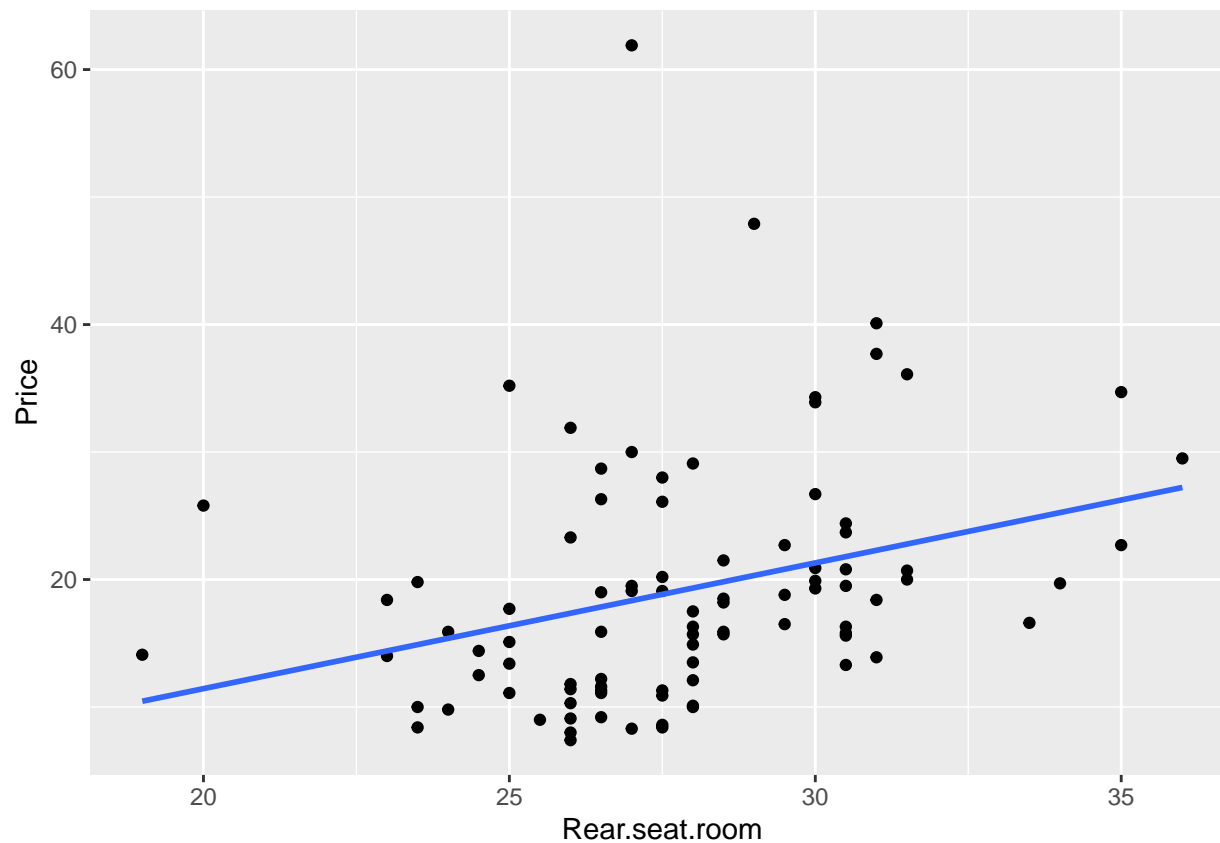


```
## [1] 0.4560279
## [1] "Rear.seat.room"

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

## Warning: Removed 2 rows containing non-finite outside the scale range
## ('stat_smooth()').

## Warning: Removed 2 rows containing missing values or values outside the scale range
## ('geom_point()').
```

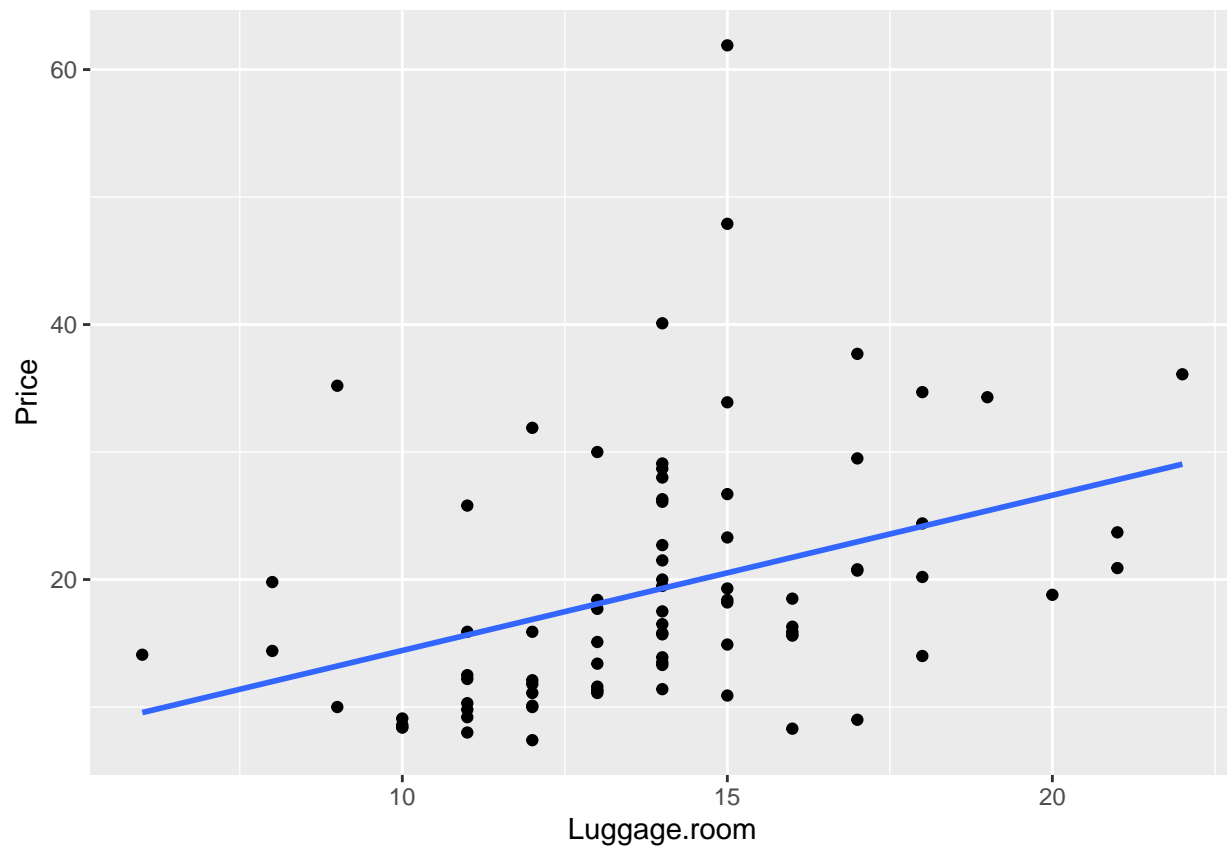



```
## [1] NA
## [1] "Luggage.room"

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

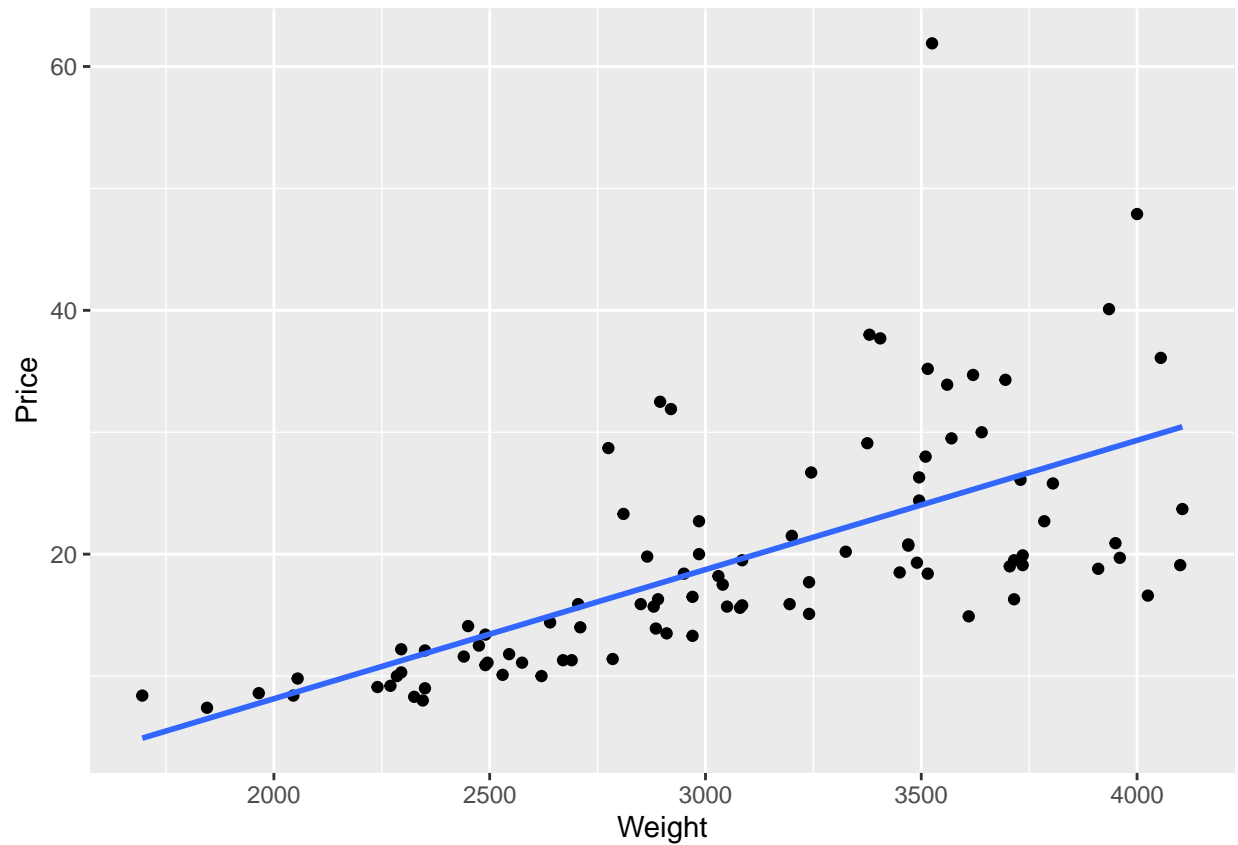
## Warning: Removed 11 rows containing non-finite outside the scale range
## ('stat_smooth()').

## Warning: Removed 11 rows containing missing values or values outside the scale range
## ('geom_point()').
```



```
## [1] NA  
## [1] "Weight"
```

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



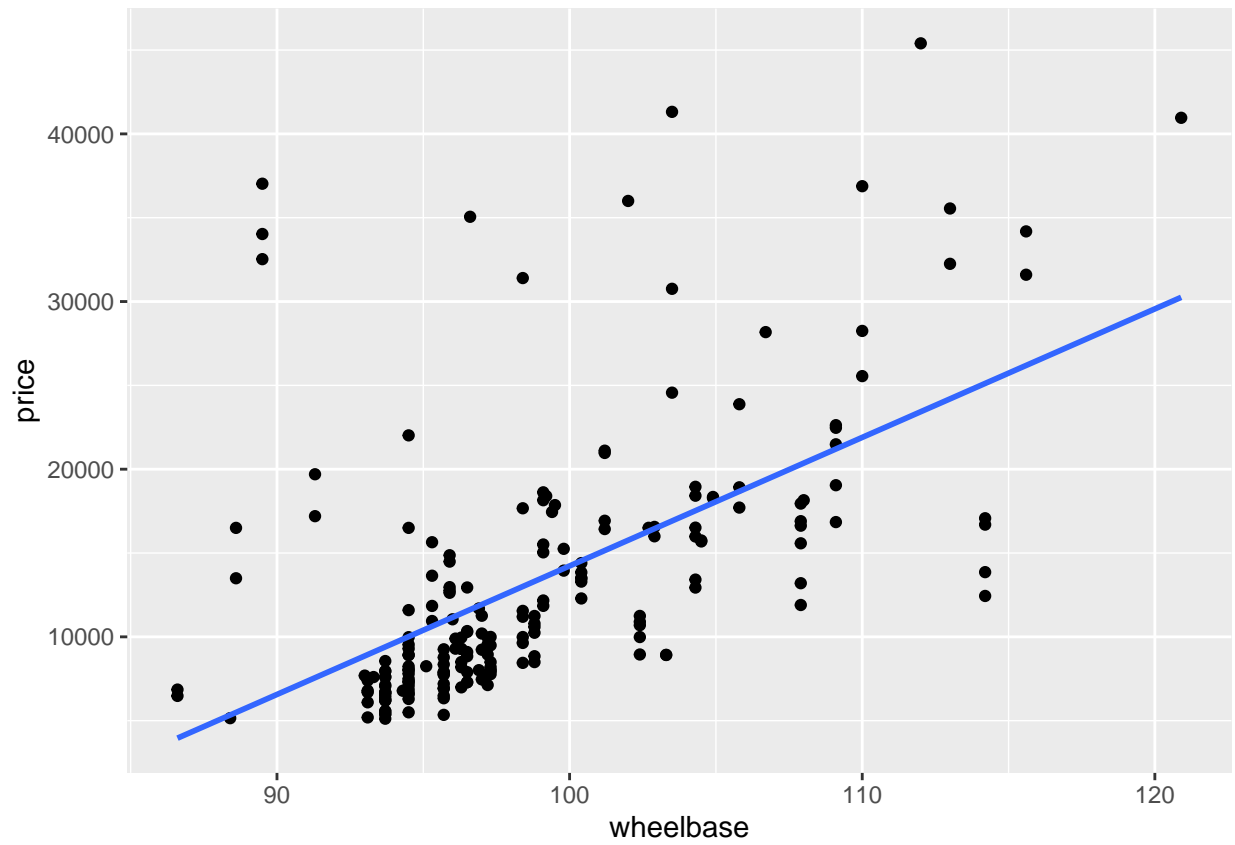
```
## [1] 0.647179
```

```
columns_assign <- c("wheelbase", "carlength", "carwidth", "carheight", "curbweight", "enginesize", "horsepower")
```

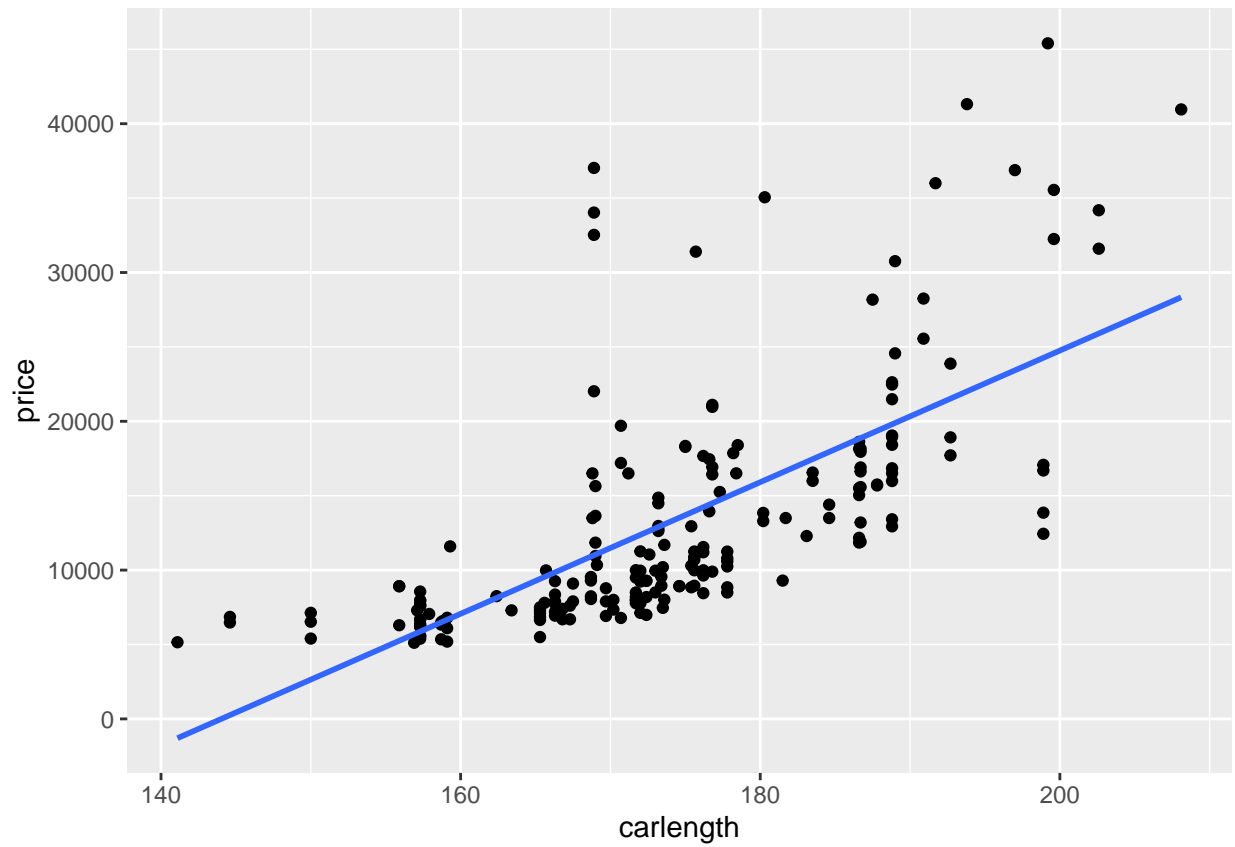
```
for (var in columns_assign)
{
  print(var);
  p <- ggplot(cars_assign, aes_string(x = var, y = "price")) +
    geom_point() +
    geom_smooth(method = "lm", se = FALSE);
  print(p);
  correlation <- cor(cars_assign$price, cars_assign[[var]]);
  print(correlation)
}
```

```
## [1] "wheelbase"
```

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

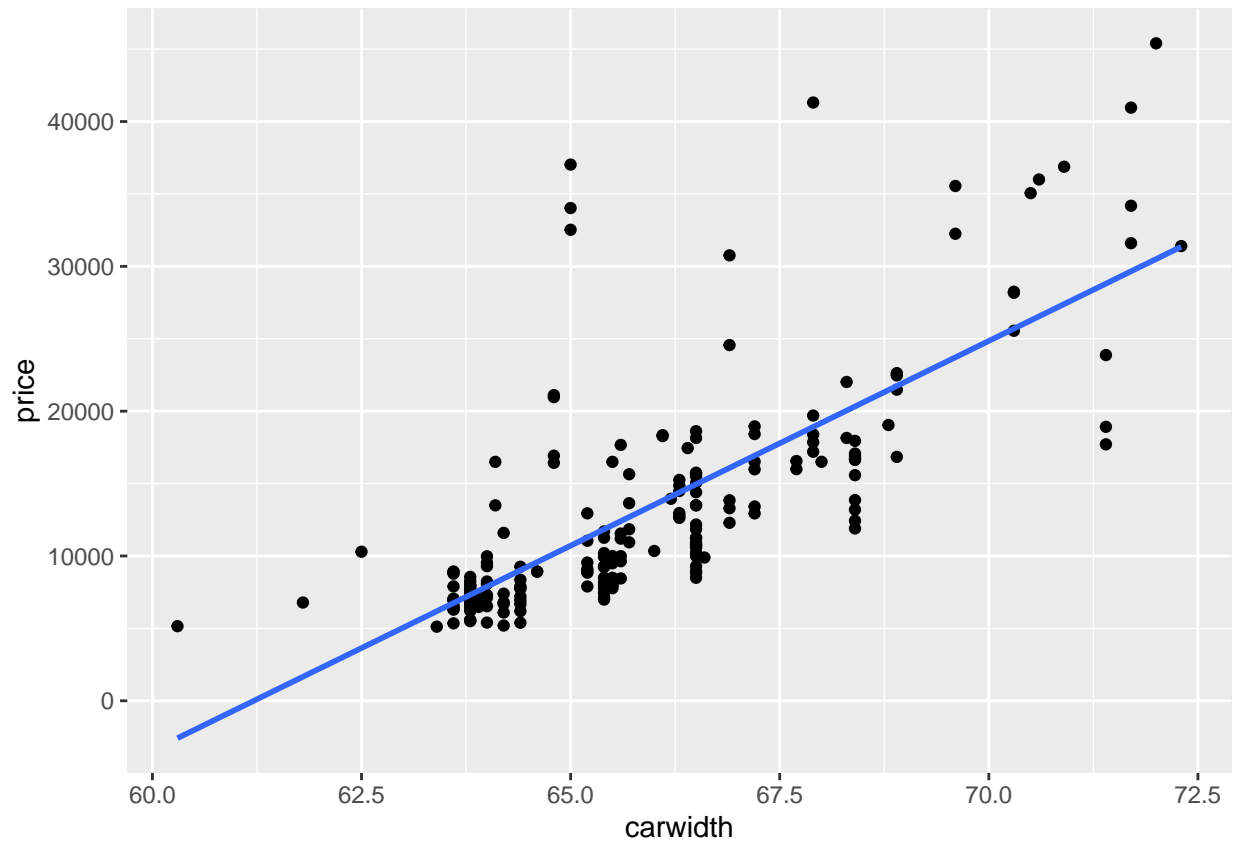


```
## [1] 0.5778156  
## [1] "carlength"  
  
## 'geom_smooth()' using formula = 'y ~ x'
```



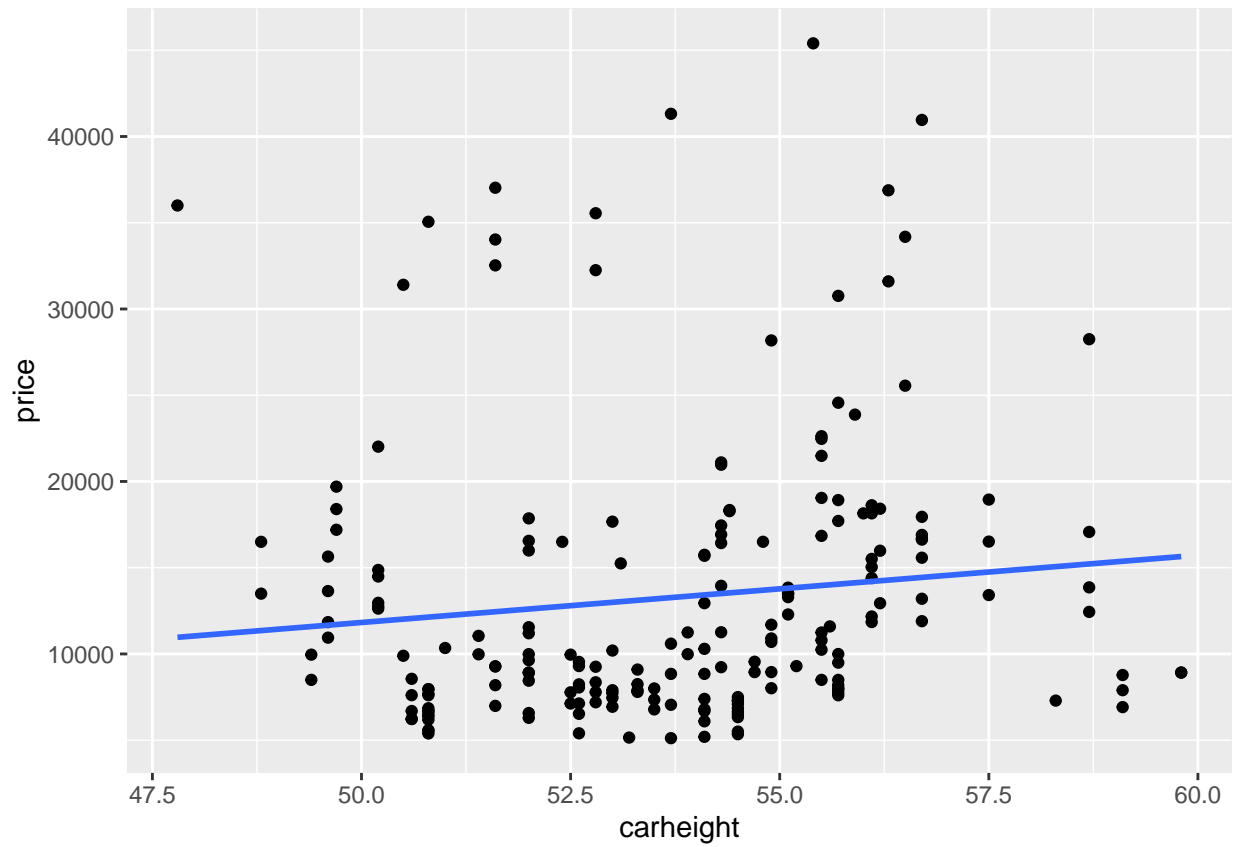
```
## [1] 0.68292  
## [1] "carwidth"
```

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



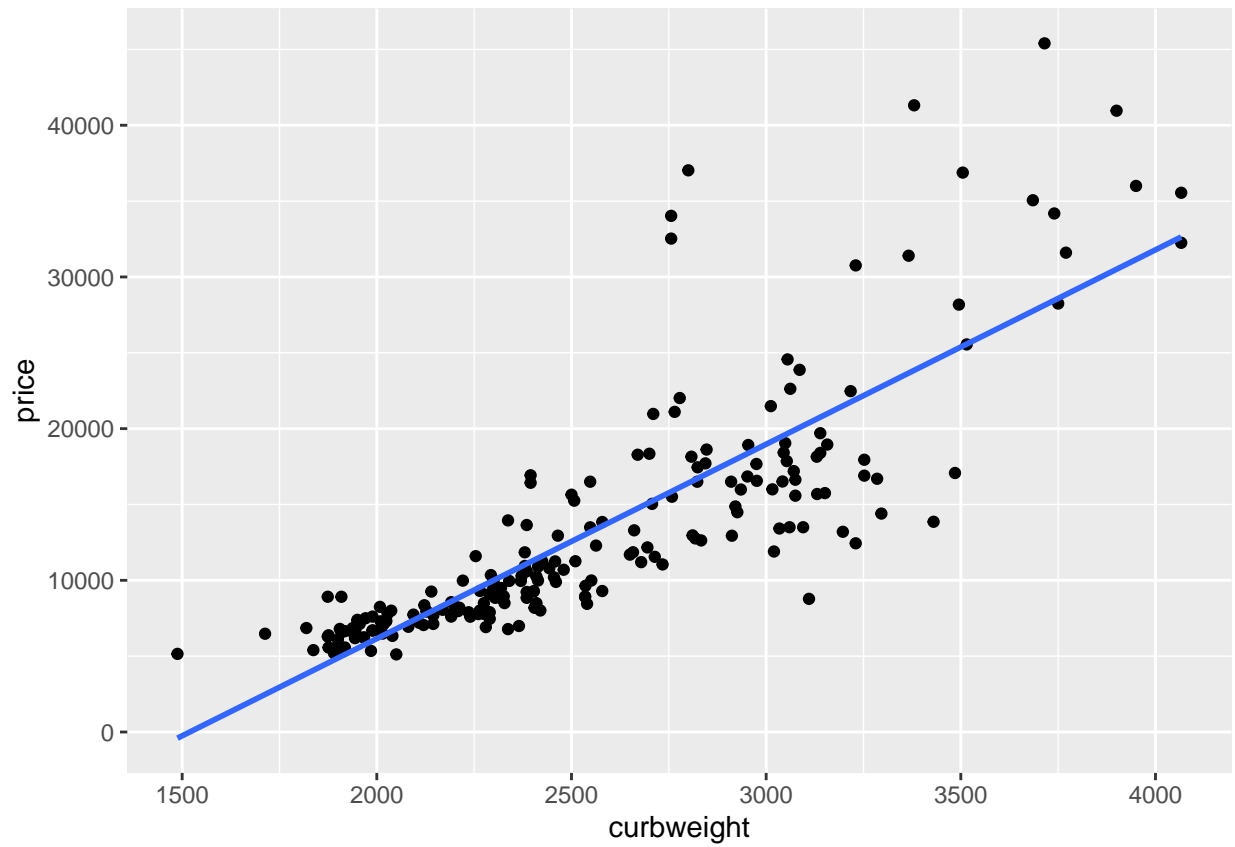
```
## [1] 0.7593253  
## [1] "carheight"
```

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



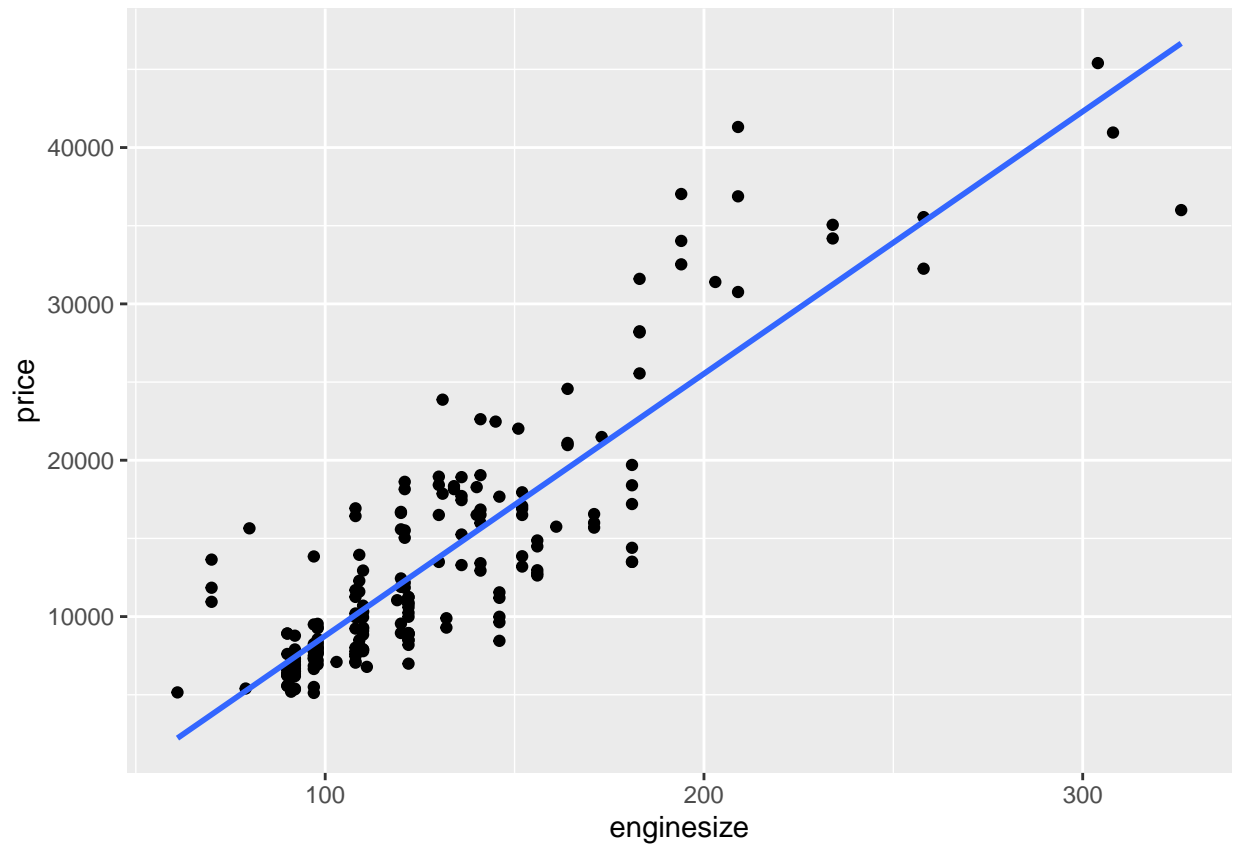
```
## [1] 0.1193362
## [1] "curbweight"

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



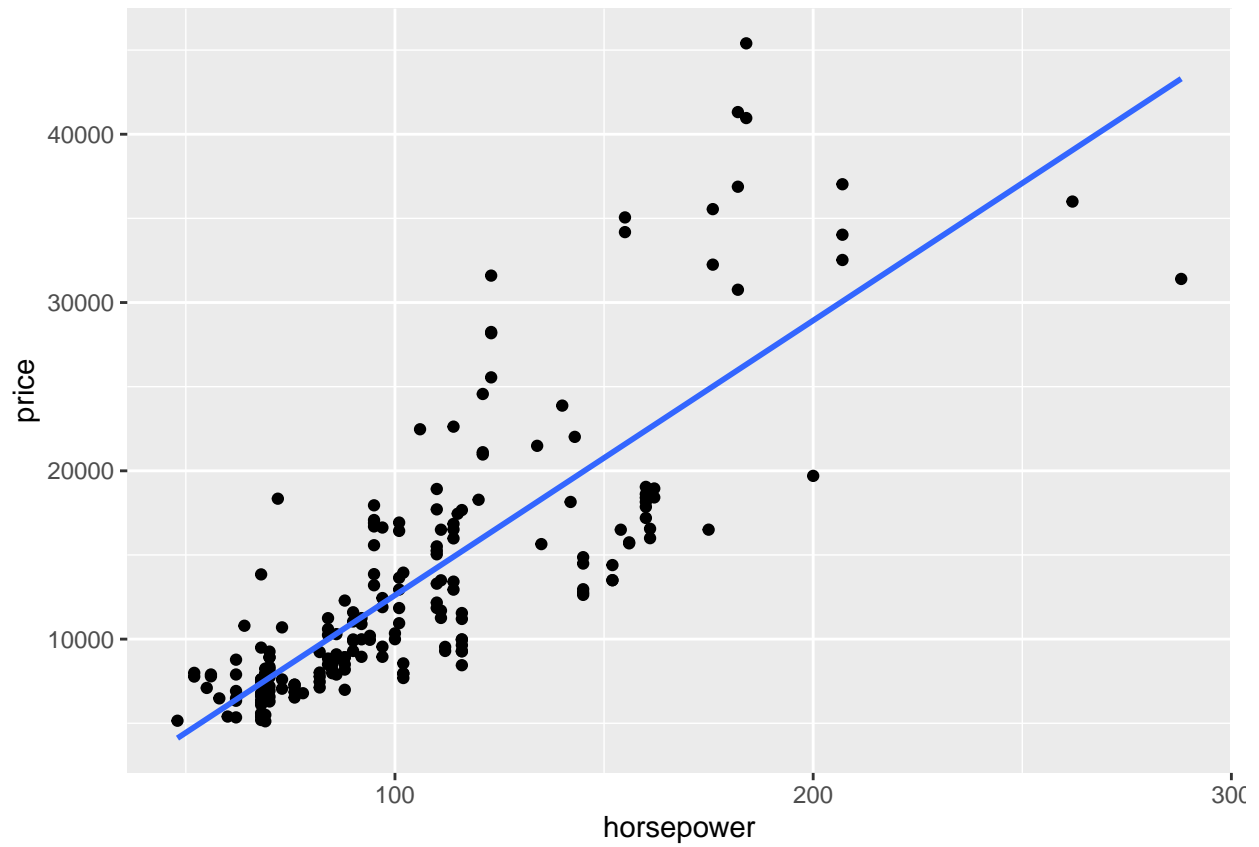
```
## [1] 0.8353049  
## [1] "enginesize"
```

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

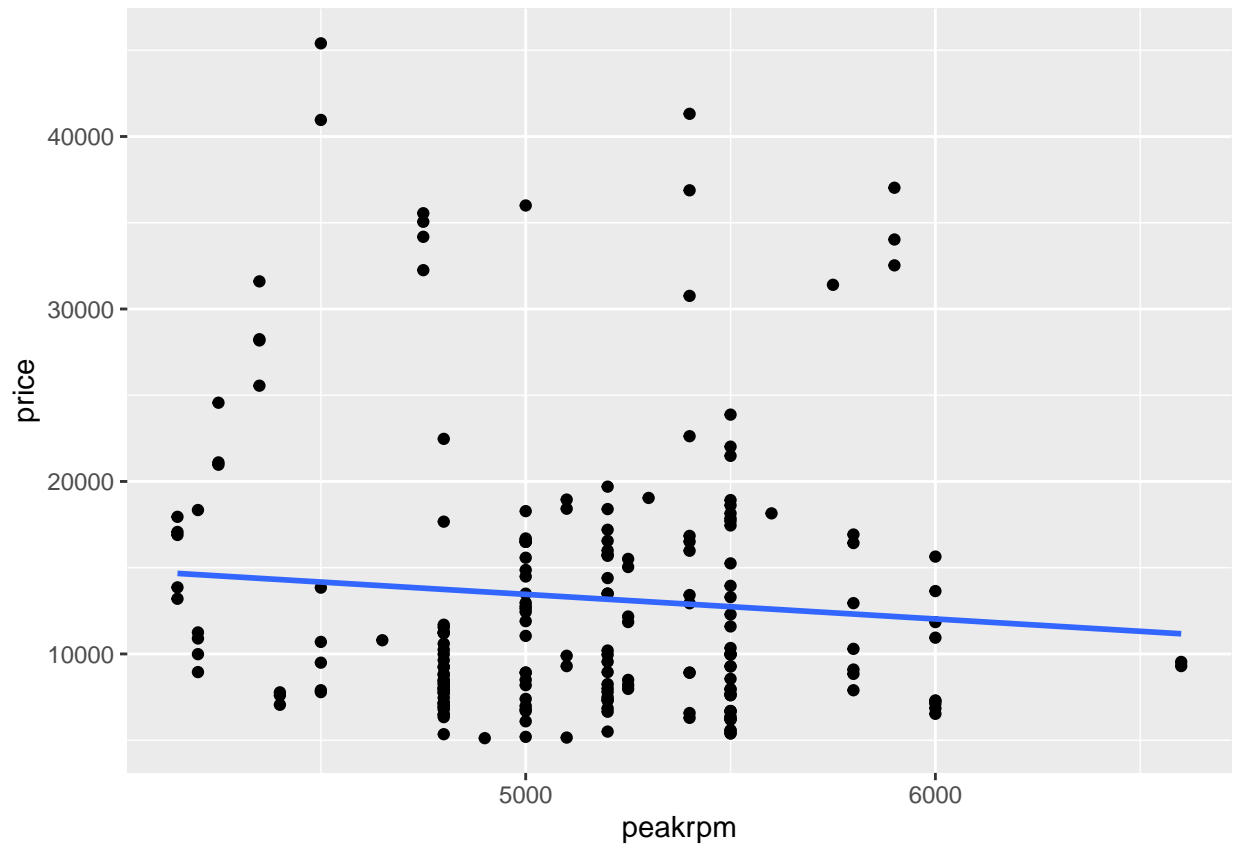
```
## [1] 0.8741448  
## [1] "horsepower"
```

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



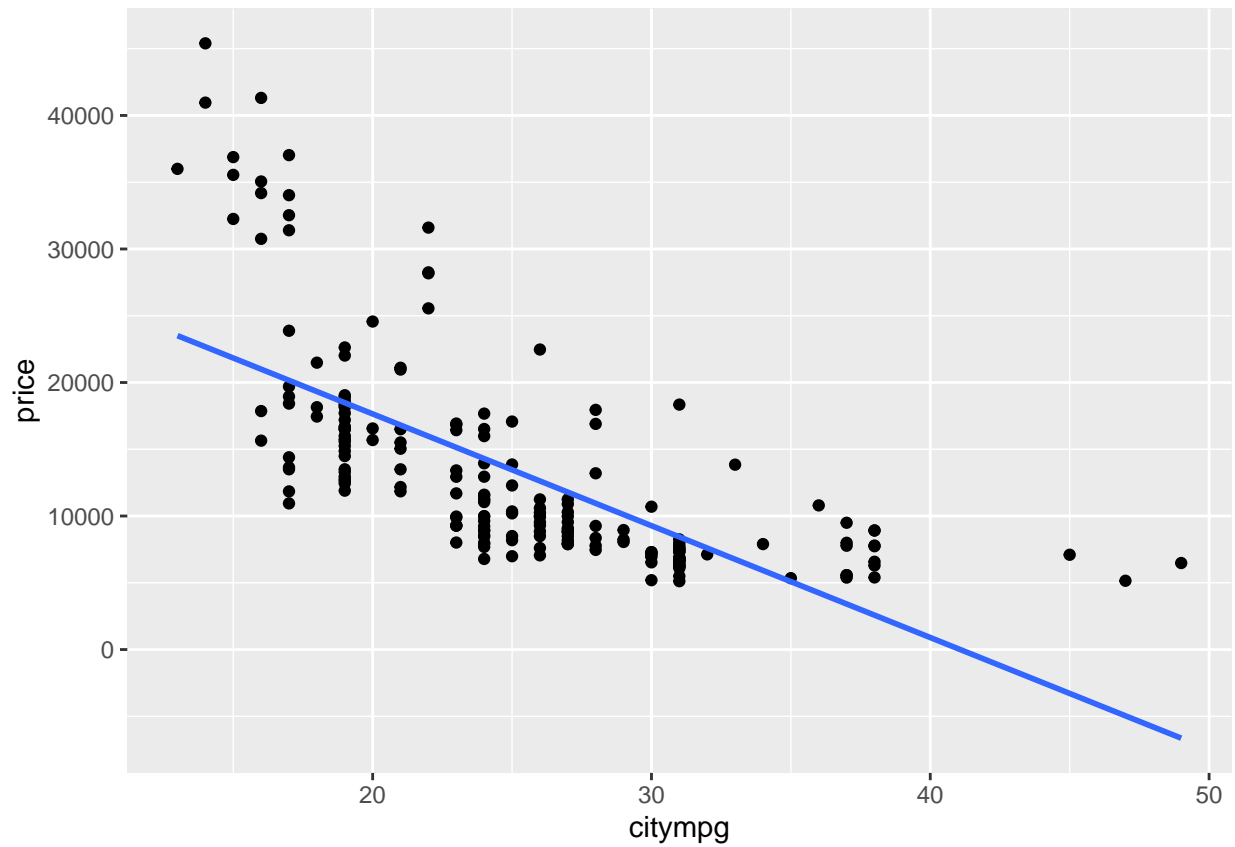
```
## [1] 0.8081388  
## [1] "peakrpm"
```

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



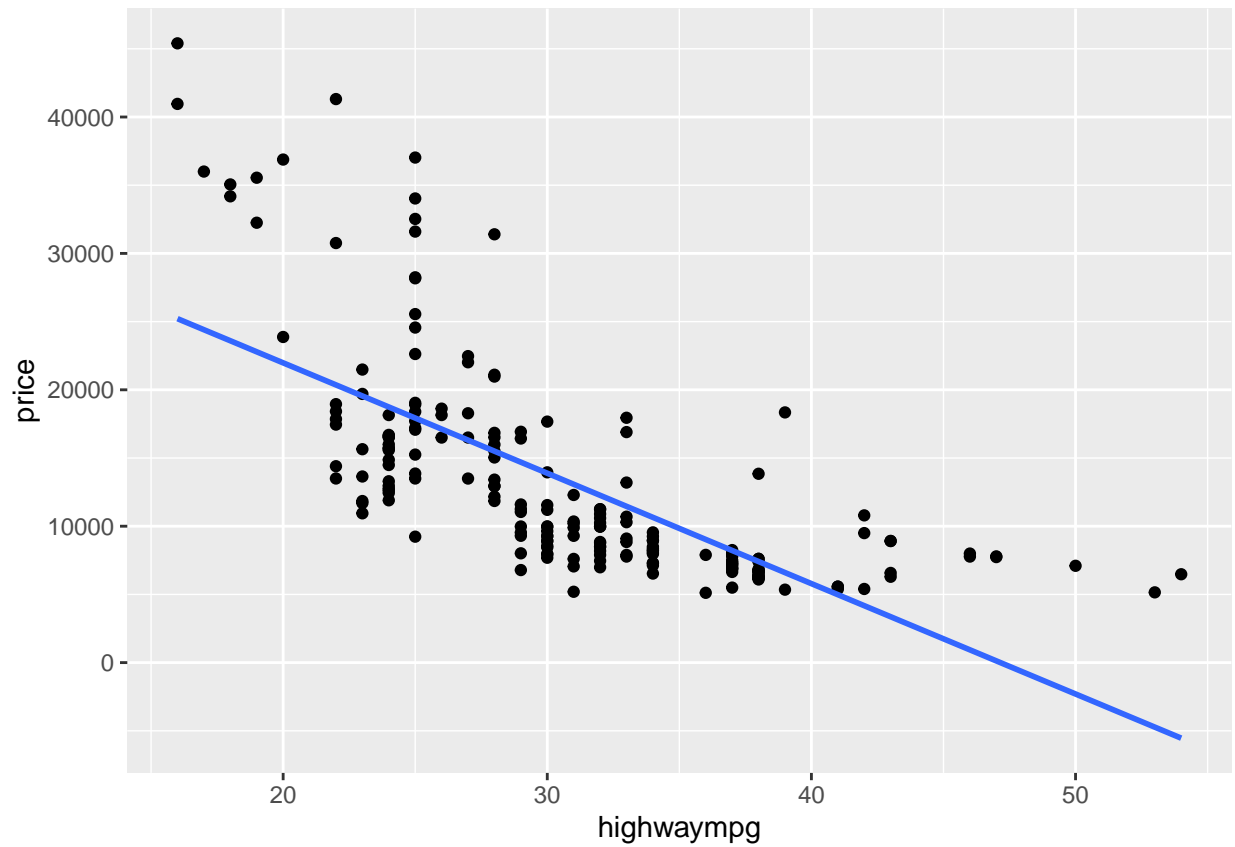
```
## [1] -0.08526715  
## [1] "citympg"
```

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



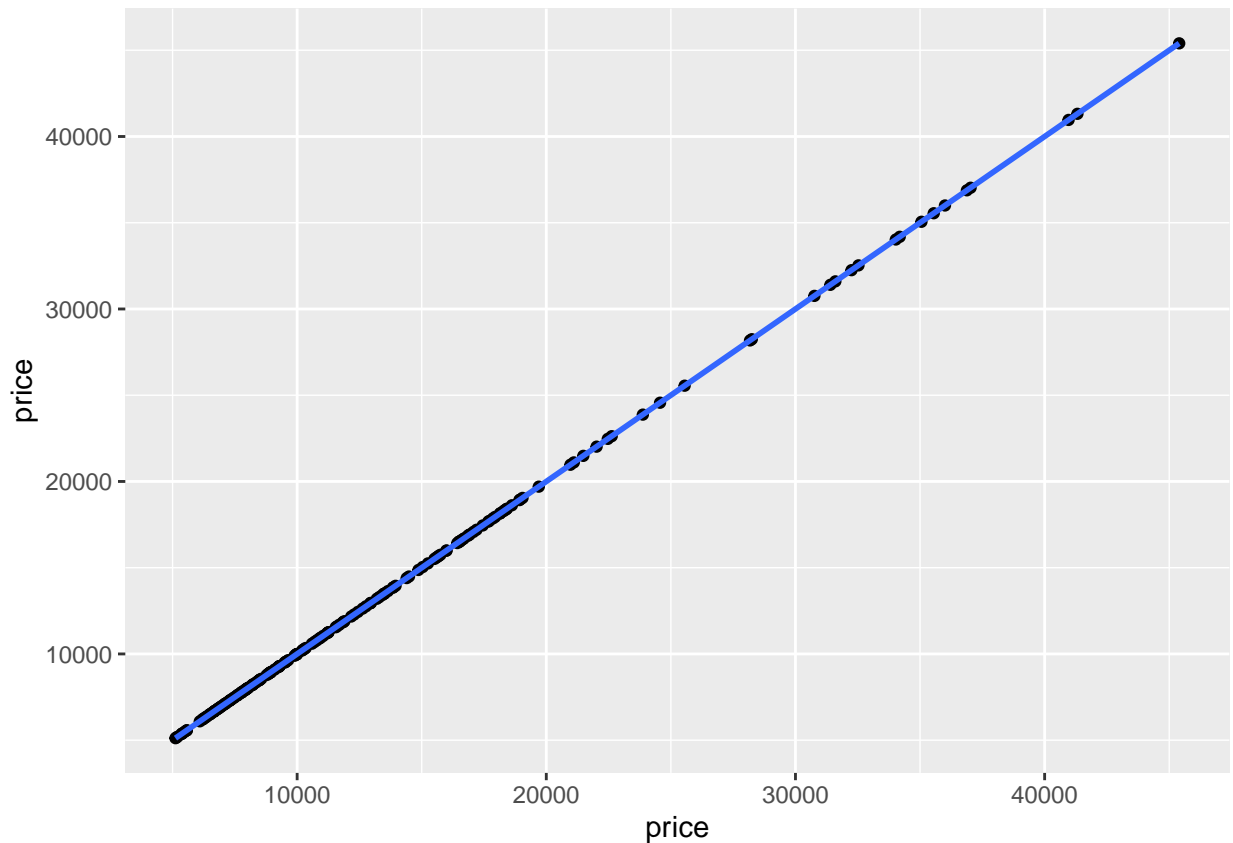
```
## [1] -0.6857513
## [1] "highwaympg"

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



```
## [1] -0.6975991  
## [1] "price"
```

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



```
## [1] 1
```

Cars93 linear regression model

```
model93 <- lm(Price ~ MPG.city + EngineSize + Horsepower + Fuel.tank.capacity + Weight, data = cars93)
summary(model93)
```

```
##
## Call:
## lm(formula = Price ~ MPG.city + EngineSize + Horsepower + Fuel.tank.capacity +
##     Weight, data = cars93)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.824  -2.713   -0.456    1.444   31.936
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -3.509880   11.026931  -0.318   0.751
## MPG.city       -0.078651    0.215438  -0.365   0.716
## EngineSize     -0.797961    1.187025  -0.672   0.503
## Horsepower      0.128456    0.019018   6.754 1.55e-09 ***
## Fuel.tank.capacity 0.040706    0.445052   0.091   0.927
## Weight         0.002523    0.003136   0.805   0.423
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.015 on 87 degrees of freedom
## Multiple R-squared:  0.6333, Adjusted R-squared:  0.6122
## F-statistic: 30.05 on 5 and 87 DF,  p-value: < 2.2e-16
```

`cars_assignment` linear regression model

```
model_assign <- lm(price ~ carwidth + curbweight + enginesize + horsepower + citympg, data = cars_assign)
summary(model_assign)
```

```
##
## Call:
## lm(formula = price ~ carwidth + curbweight + enginesize + horsepower +
##     citympg, data = cars_assign)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -8184.4 -1715.3   -91.6  1339.2 14273.7
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -47388.322  13175.253  -3.597 0.000406 ***
## carwidth      574.857    225.058   2.554 0.011390 *
## curbweight     2.378      1.324   1.795 0.074101 .
## enginesize     82.921     13.342   6.215 2.95e-09 ***
## horsepower     52.703     13.280   3.968 0.000101 ***
## citympg        27.400      70.724   0.387 0.698860
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3432 on 199 degrees of freedom
## Multiple R-squared:  0.8199, Adjusted R-squared:  0.8154
## F-statistic: 181.2 on 5 and 199 DF,  p-value: < 2.2e-16
```

`mpgcity+highway horsepower enginesize weight`

`cars_assign` - Correlation between pairs of variables except price

```
for (var in c("carwidth", "curbweight", "enginesize", "horsepower", "citympg")) {
  for (var1 in c("carwidth", "curbweight", "enginesize", "horsepower", "citympg")) {

    if (var != var1) {
      print(paste("x =", var, ", y =", var1))

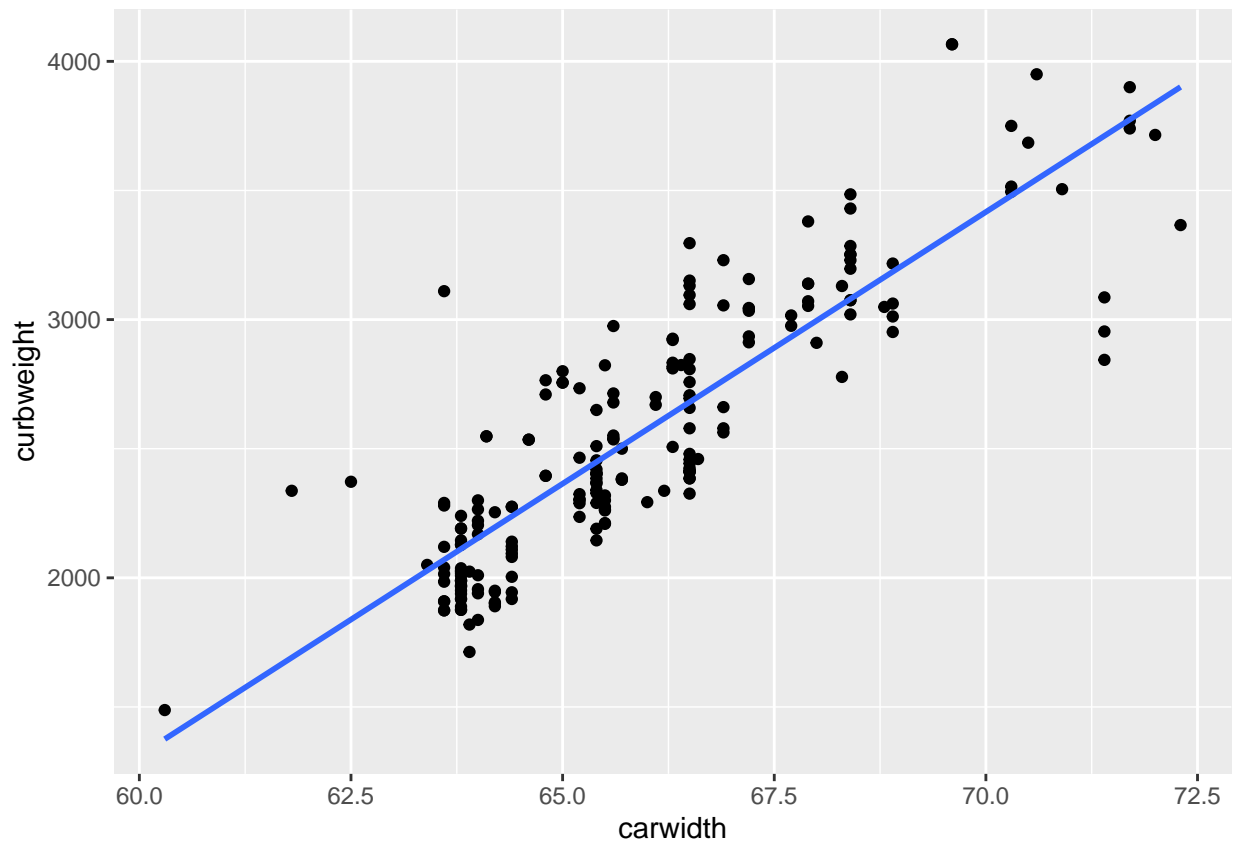
      p <- ggplot(cars_assign, aes_string(x = var, y = var1)) +
        geom_point() +
        geom_smooth(method = "lm", se = FALSE)
      print(p)

      correlation <- cor(cars_assign[[var1]], cars_assign[[var]])
      print(paste("Correlation between", var, "and", var1, ":", correlation))
    }
  }
}
```

```
}  
}
```

```
## [1] "x = carwidth , y = curbweight"
```

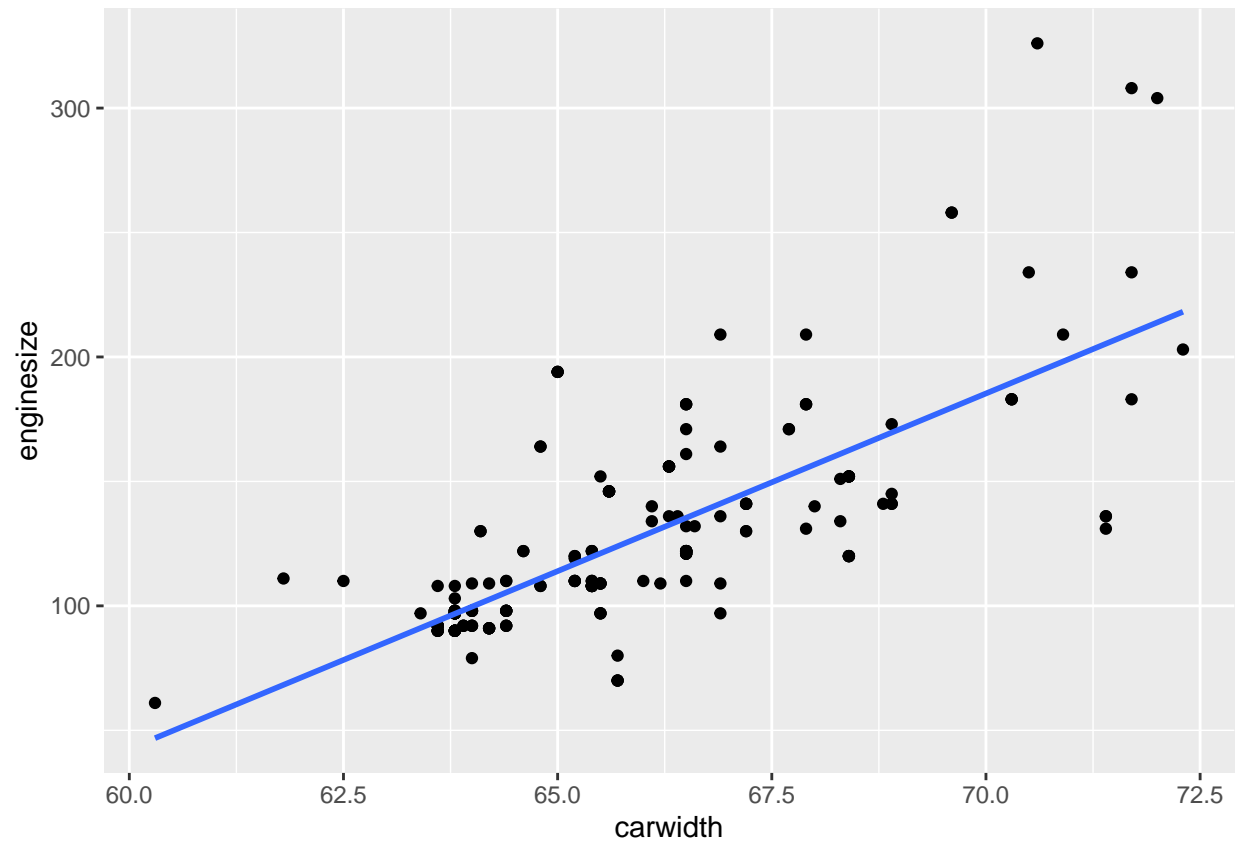
```
## 'geom_smooth()' using formula = 'y ~ x'
```



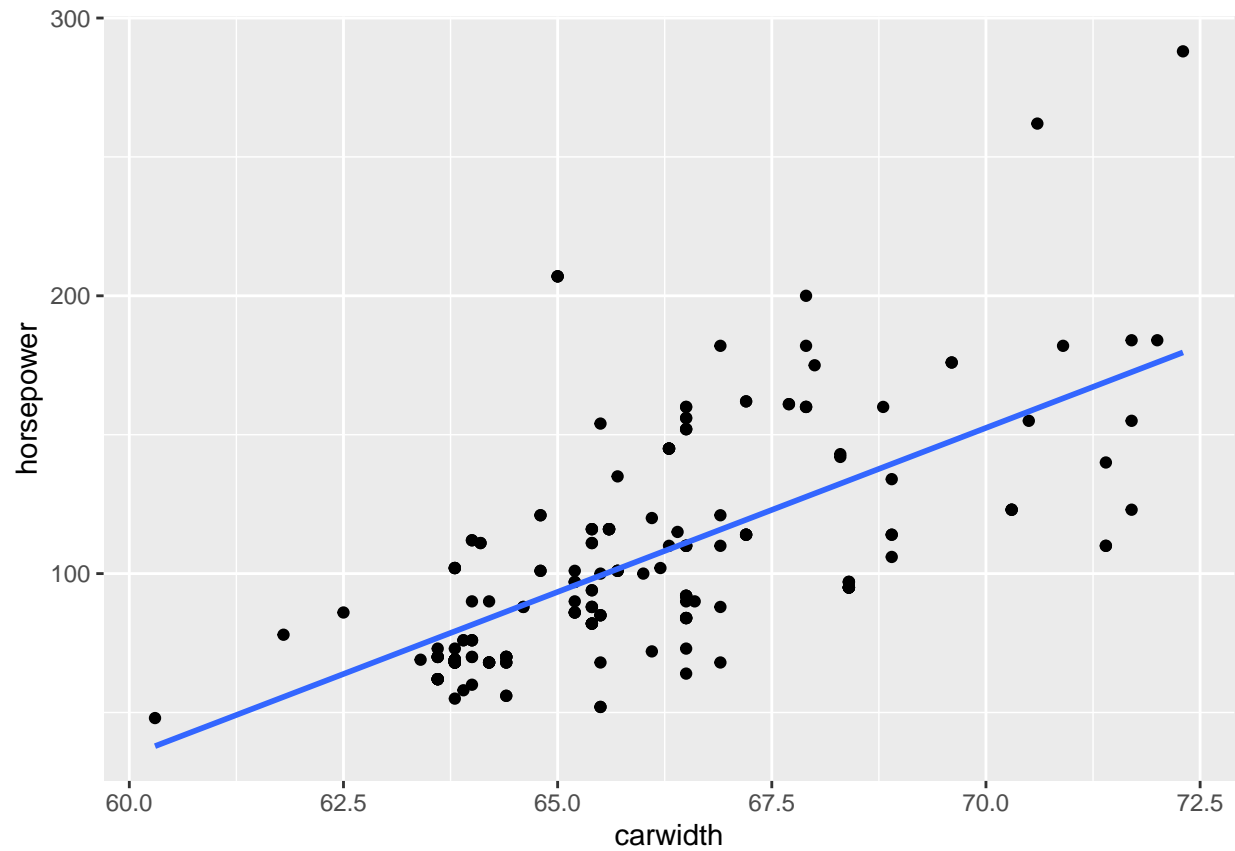
```
## [1] "Correlation between carwidth and curbweight : 0.867032464679123"
```

```
## [1] "x = carwidth , y = enginesize"
```

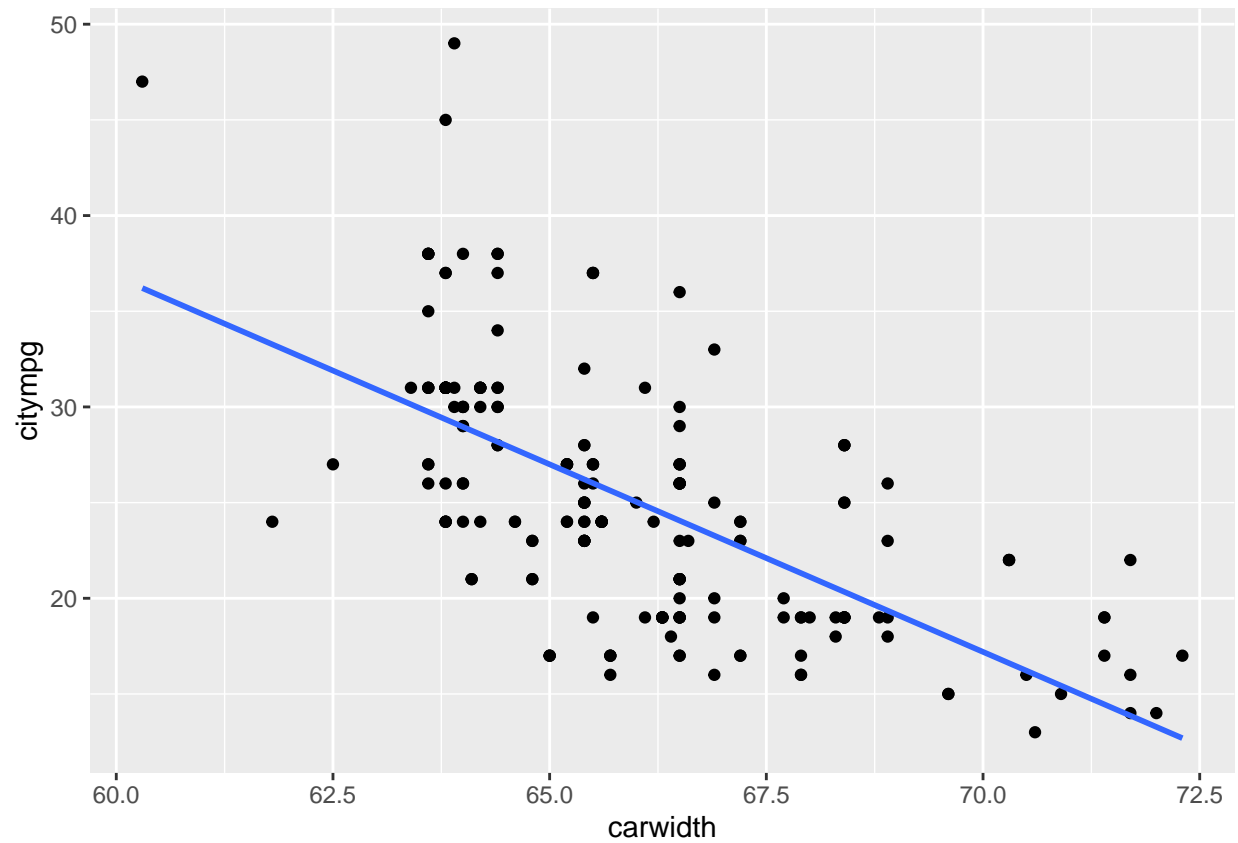
```
## 'geom_smooth()' using formula = 'y ~ x'
```

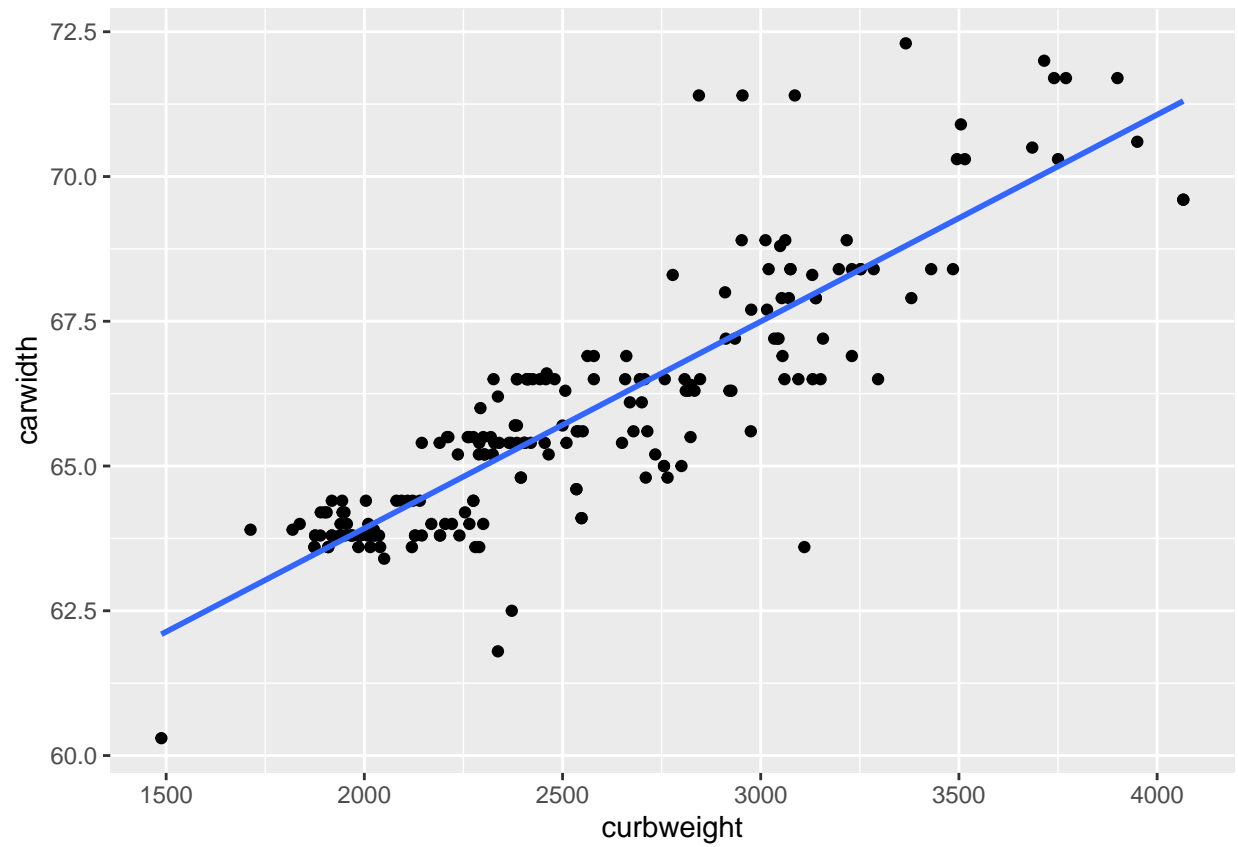
```
## [1] "Correlation between carwidth and enginesize : 0.735433404779637"  
## [1] "x = carwidth , y = horsepower"  
  
## 'geom_smooth()' using formula = 'y ~ x'
```



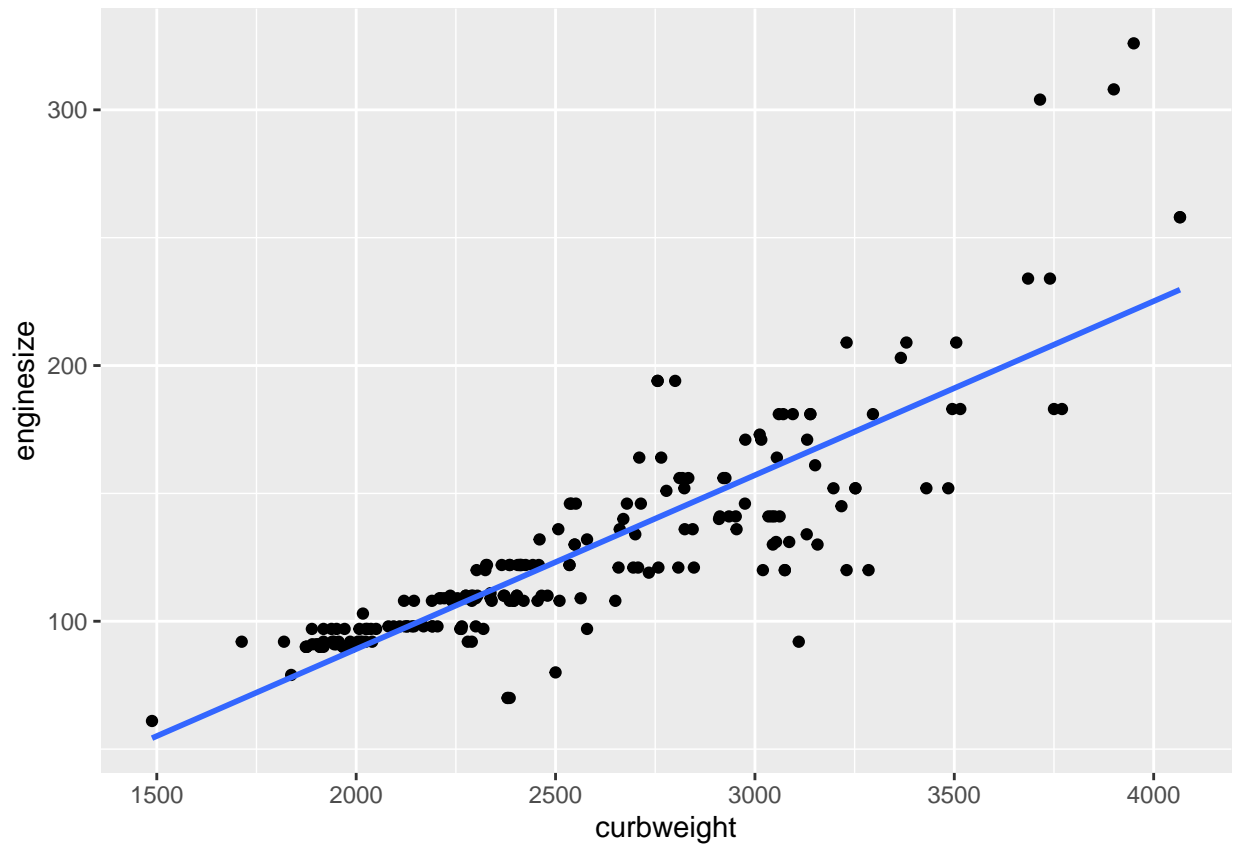
```
## [1] "Correlation between carwidth and horsepower : 0.640732075572292"  
## [1] "x = carwidth , y = citympg"  
  
## 'geom_smooth()' using formula = 'y ~ x'
```



```
## [1] "Correlation between carwidth and citympg : -0.642704340710898"  
## [1] "x = curbweight , y = carwidth"  
  
## 'geom_smooth()' using formula = 'y ~ x'
```

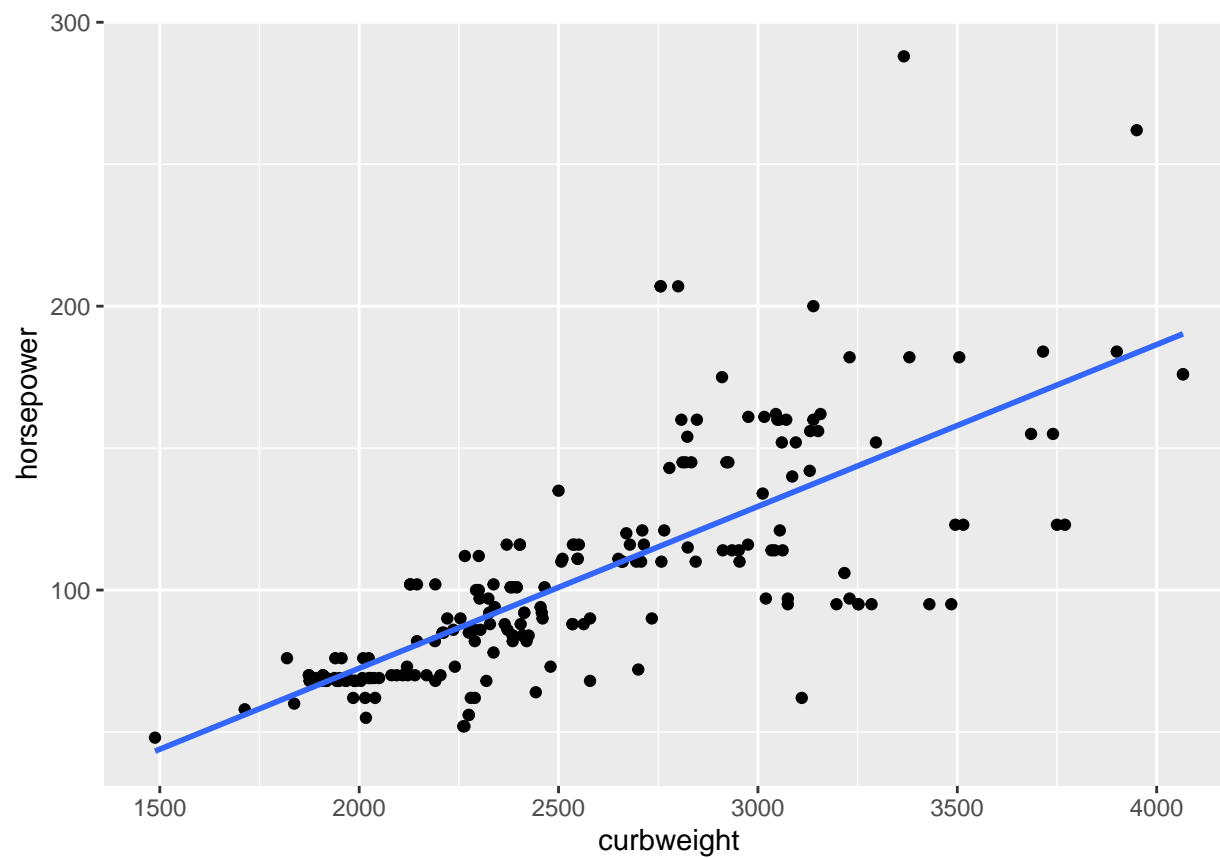


```
## [1] "Correlation between curbweight and carwidth : 0.867032464679123"  
## [1] "x = curbweight , y = enginesize"  
  
## 'geom_smooth()' using formula = 'y ~ x'
```



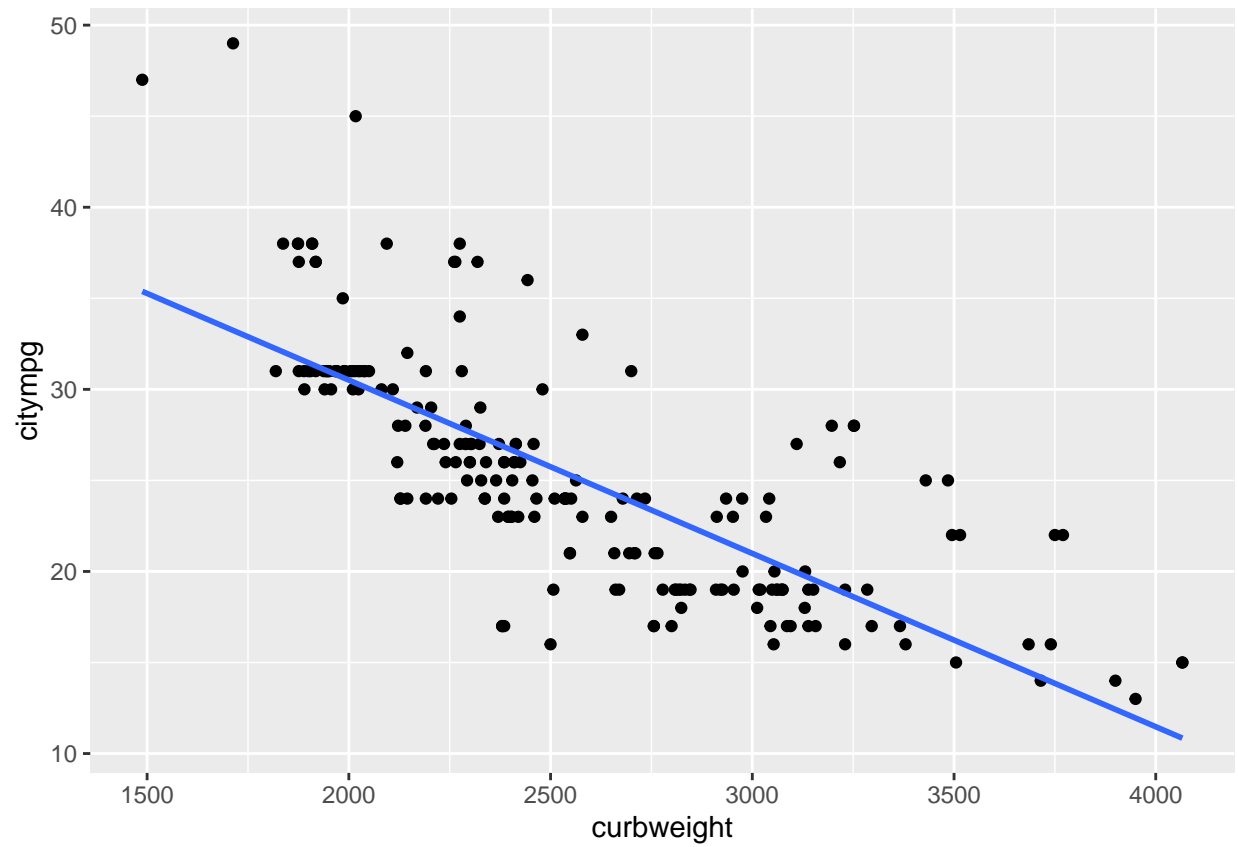
```
## [1] "Correlation between curbweight and enginesize : 0.850594073426277"  
## [1] "x = curbweight , y = horsepower"
```

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

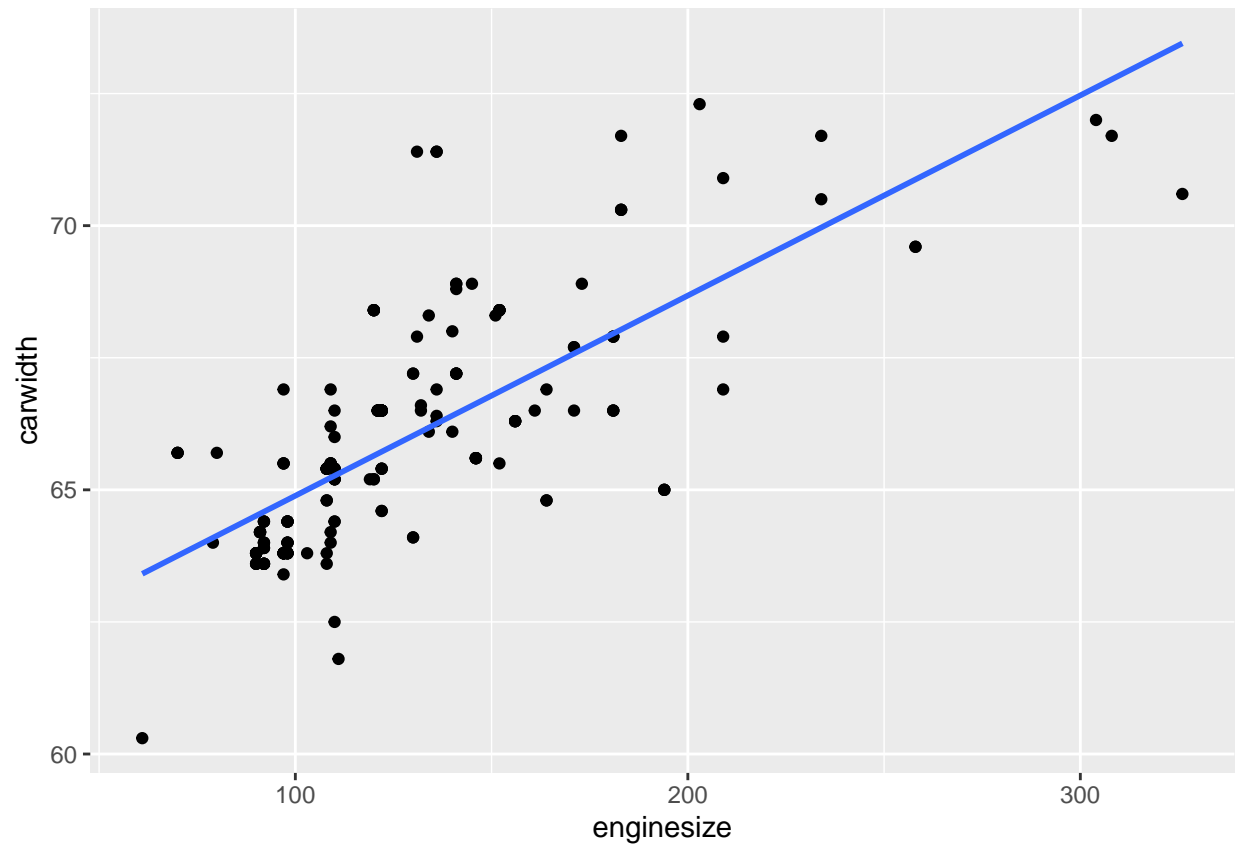


```
## [1] "Correlation between curbweight and horsepower : 0.750739251443481"  
## [1] "x = curbweight , y = citympg"
```

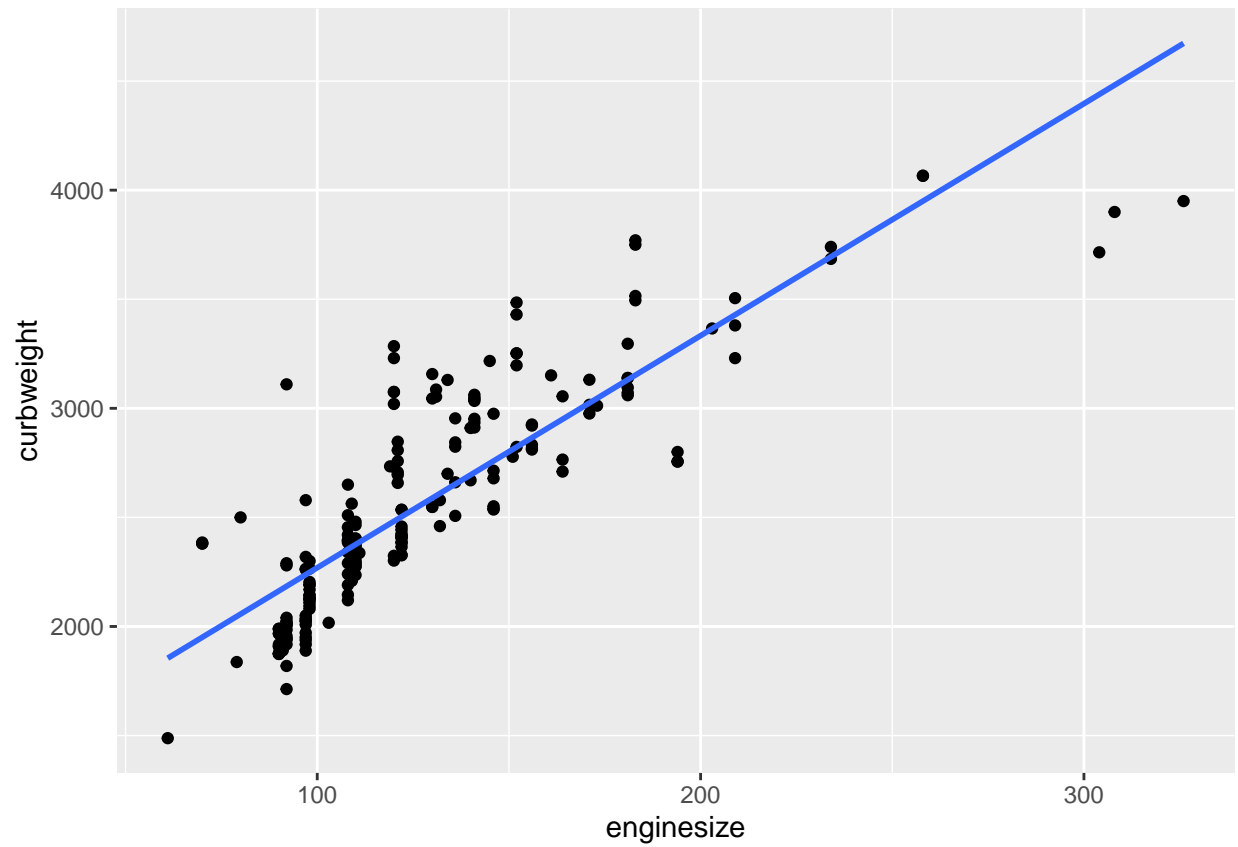
```
## 'geom_smooth()' using formula = 'y ~ x'
```



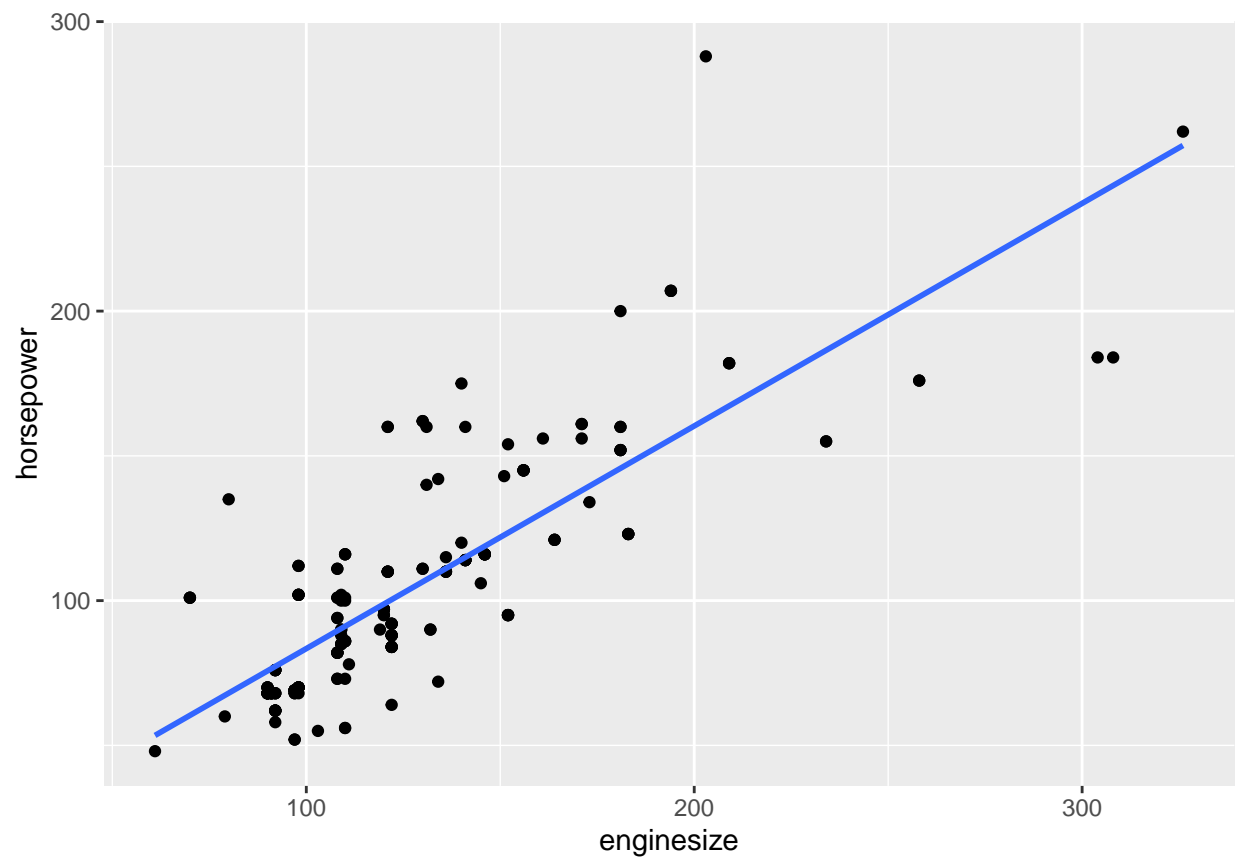
```
## [1] "Correlation between curbweight and citympg : -0.757413784505601"  
## [1] "x = enginesize , y = carwidth"  
  
## 'geom_smooth()' using formula = 'y ~ x'
```



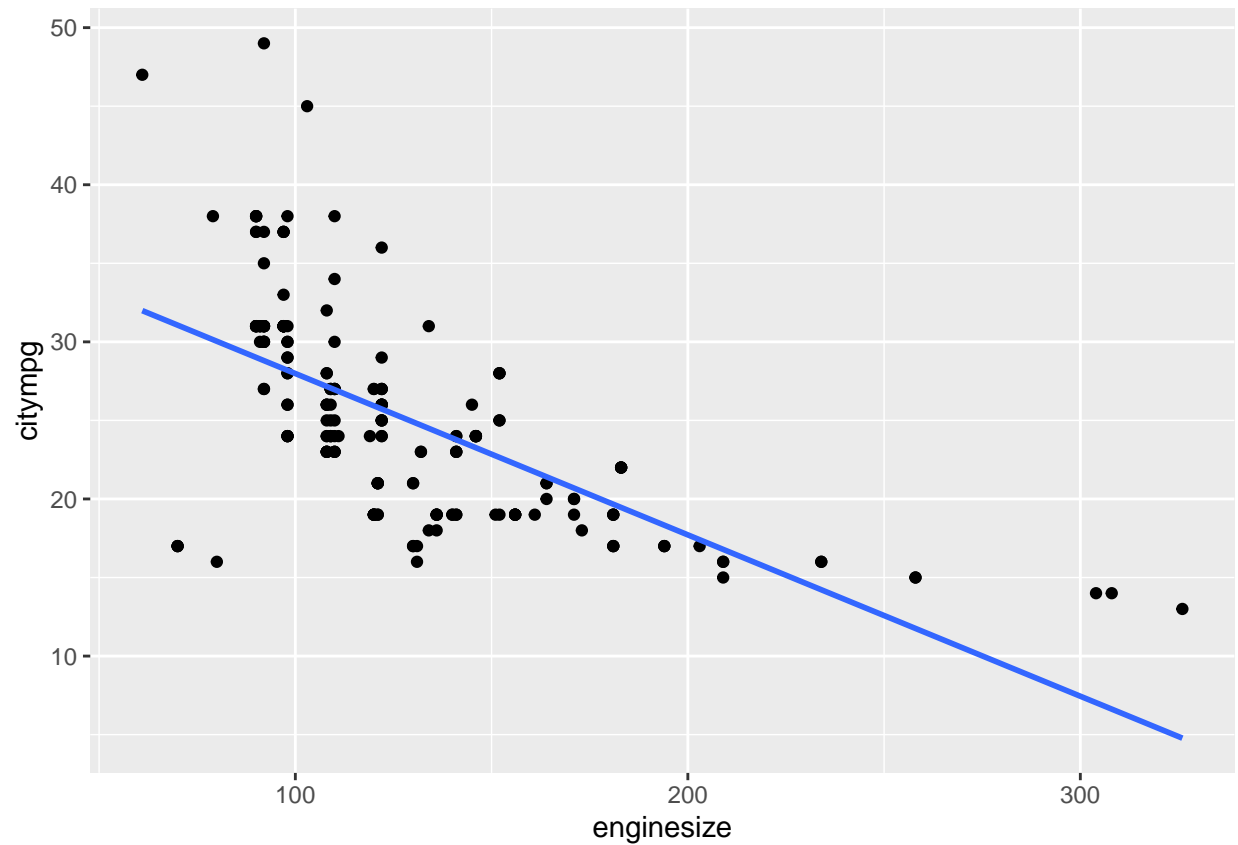
```
## [1] "Correlation between enginesize and carwidth : 0.735433404779637"  
## [1] "x = enginesize , y = curbweight"  
  
## 'geom_smooth()' using formula = 'y ~ x'
```

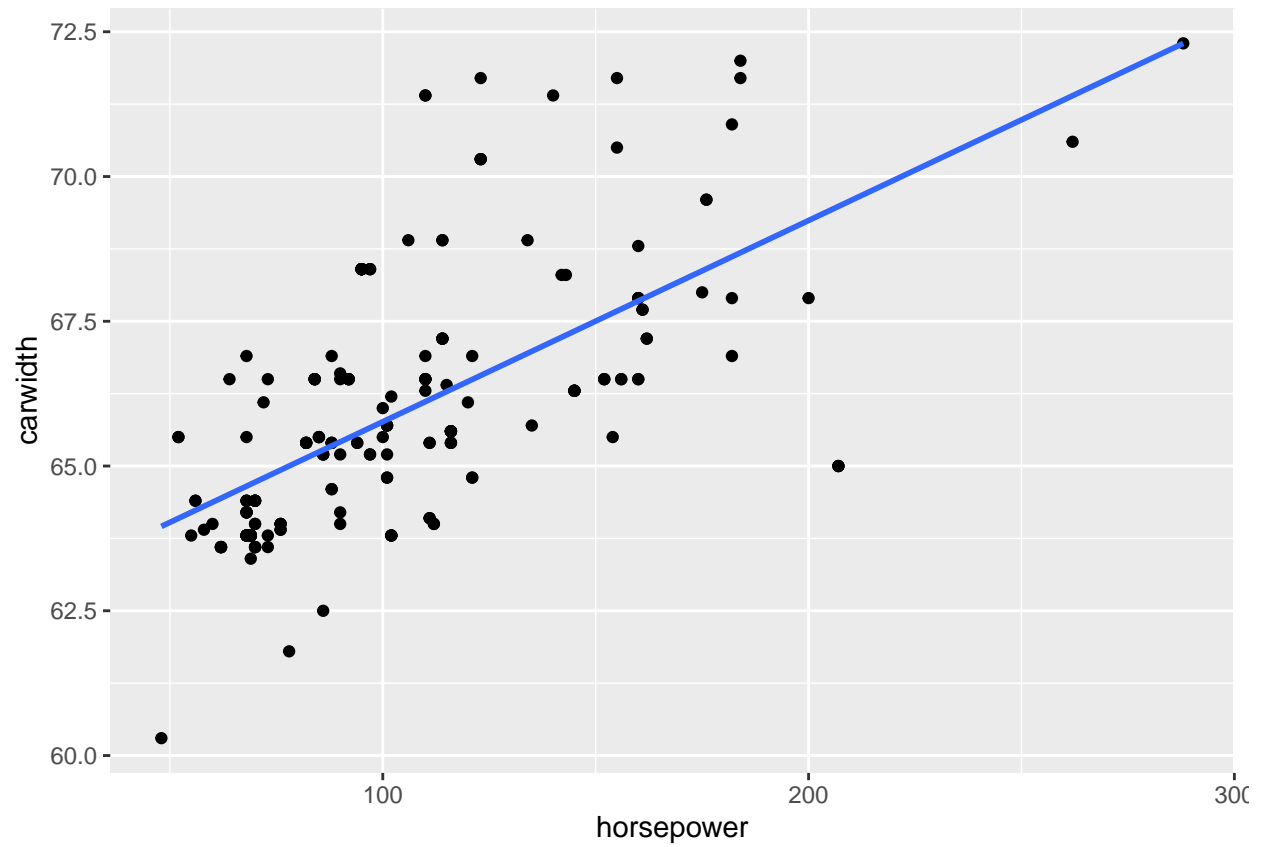
```
## [1] "Correlation between enginesize and curbweight : 0.850594073426277"  
## [1] "x = enginesize , y = horsepower"  
  
## 'geom_smooth()' using formula = 'y ~ x'
```



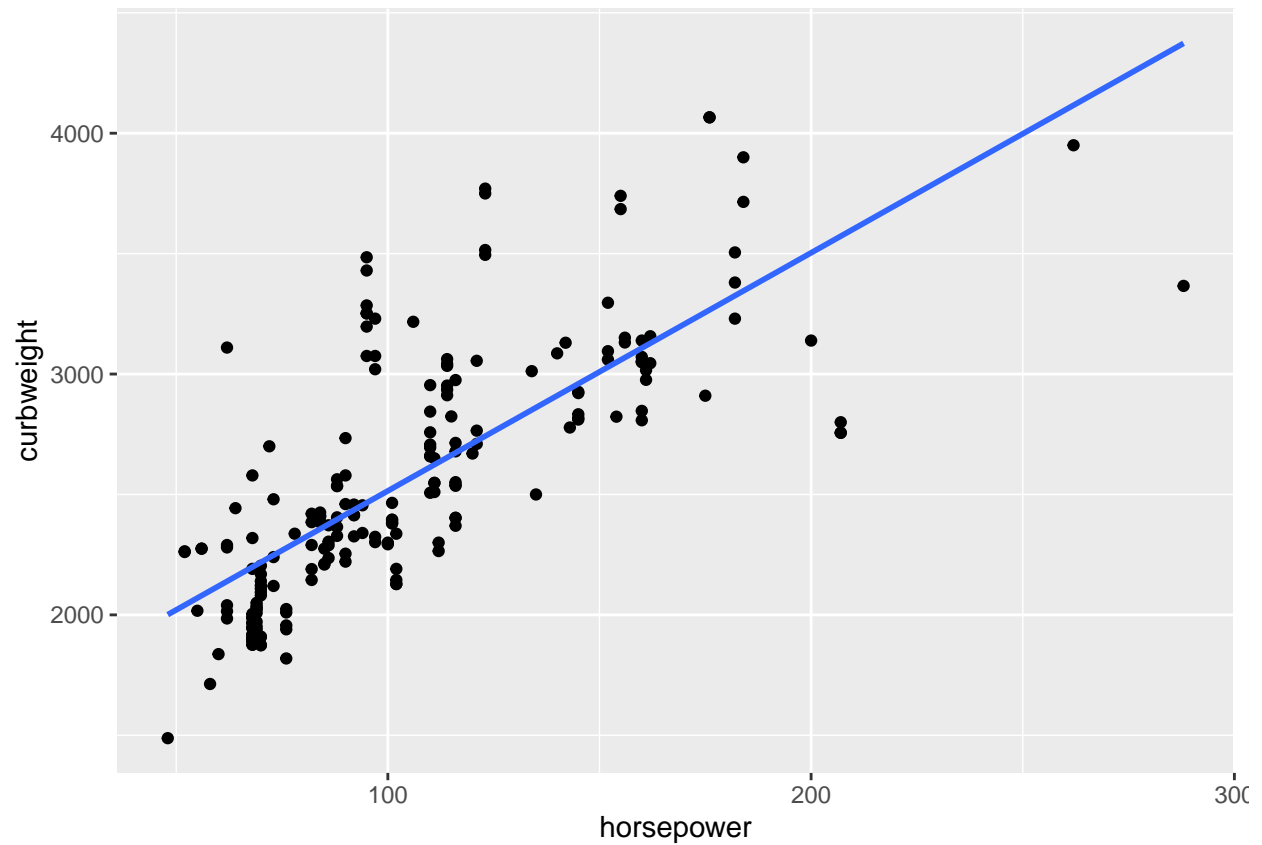
```
## [1] "Correlation between enginesize and horsepower : 0.80976865453773"  
## [1] "x = enginesize , y = citympg"  
  
## 'geom_smooth()' using formula = 'y ~ x'
```



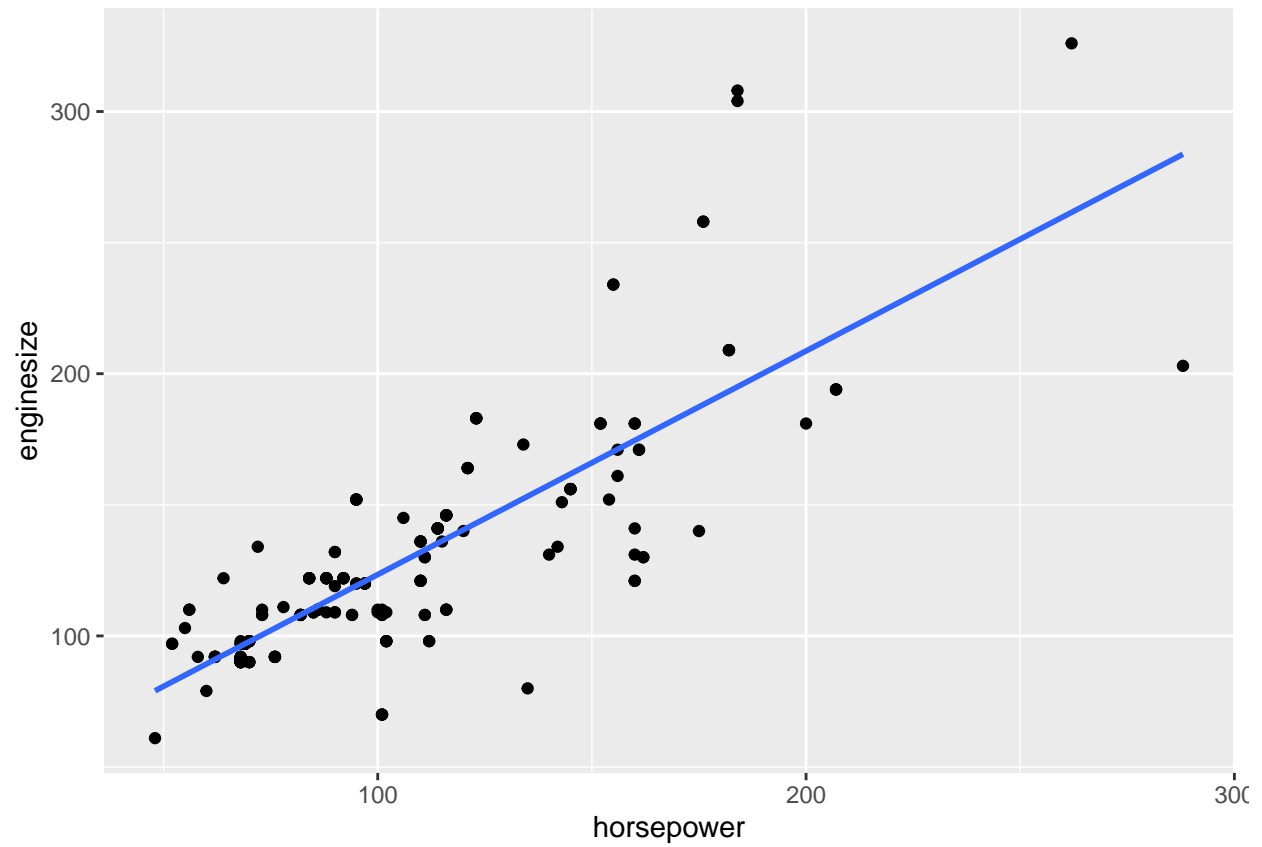
```
## [1] "Correlation between enginesize and citympg : -0.65365791631142"  
## [1] "x = horsepower , y = carwidth"  
  
## 'geom_smooth()' using formula = 'y ~ x'
```



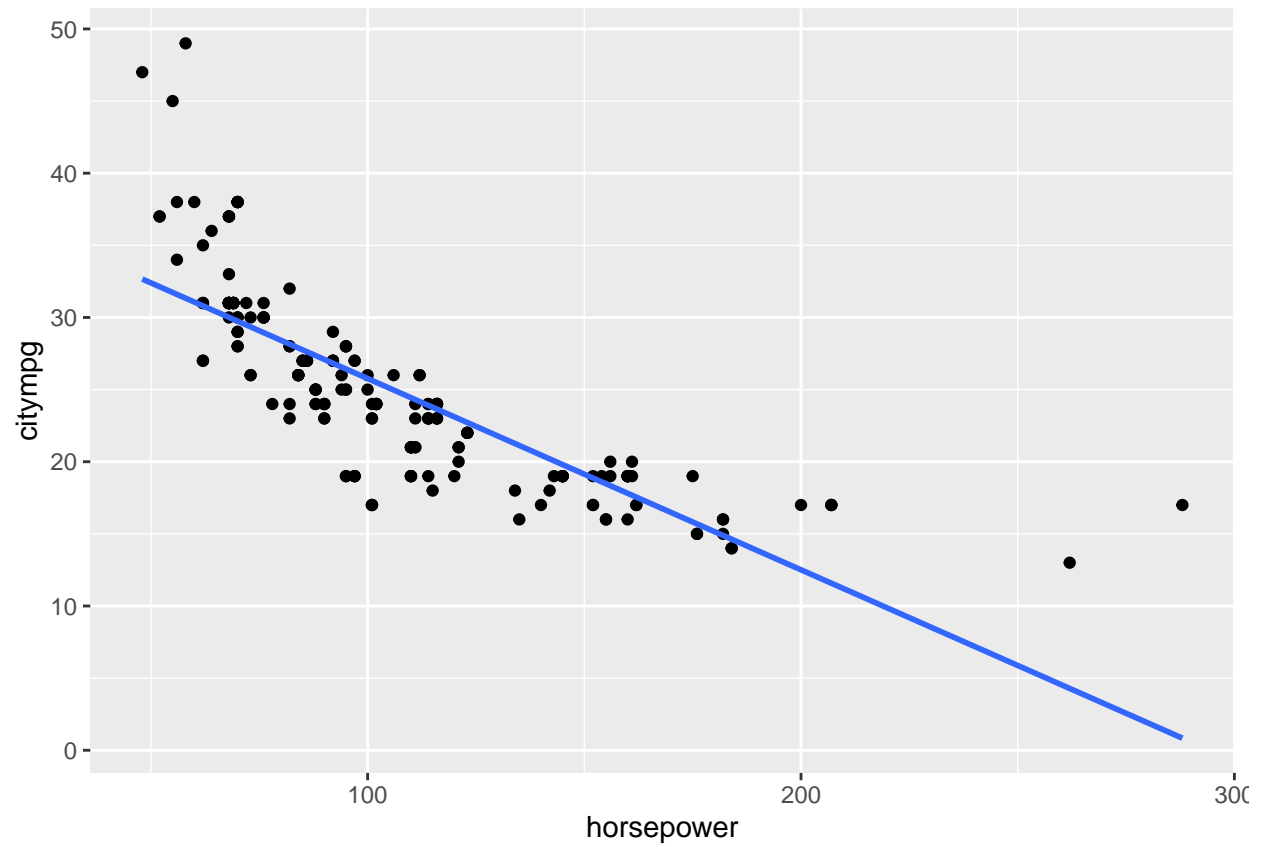
```
## [1] "Correlation between horsepower and carwidth : 0.640732075572292"  
## [1] "x = horsepower , y = curbweight"  
  
## 'geom_smooth()' using formula = 'y ~ x'
```



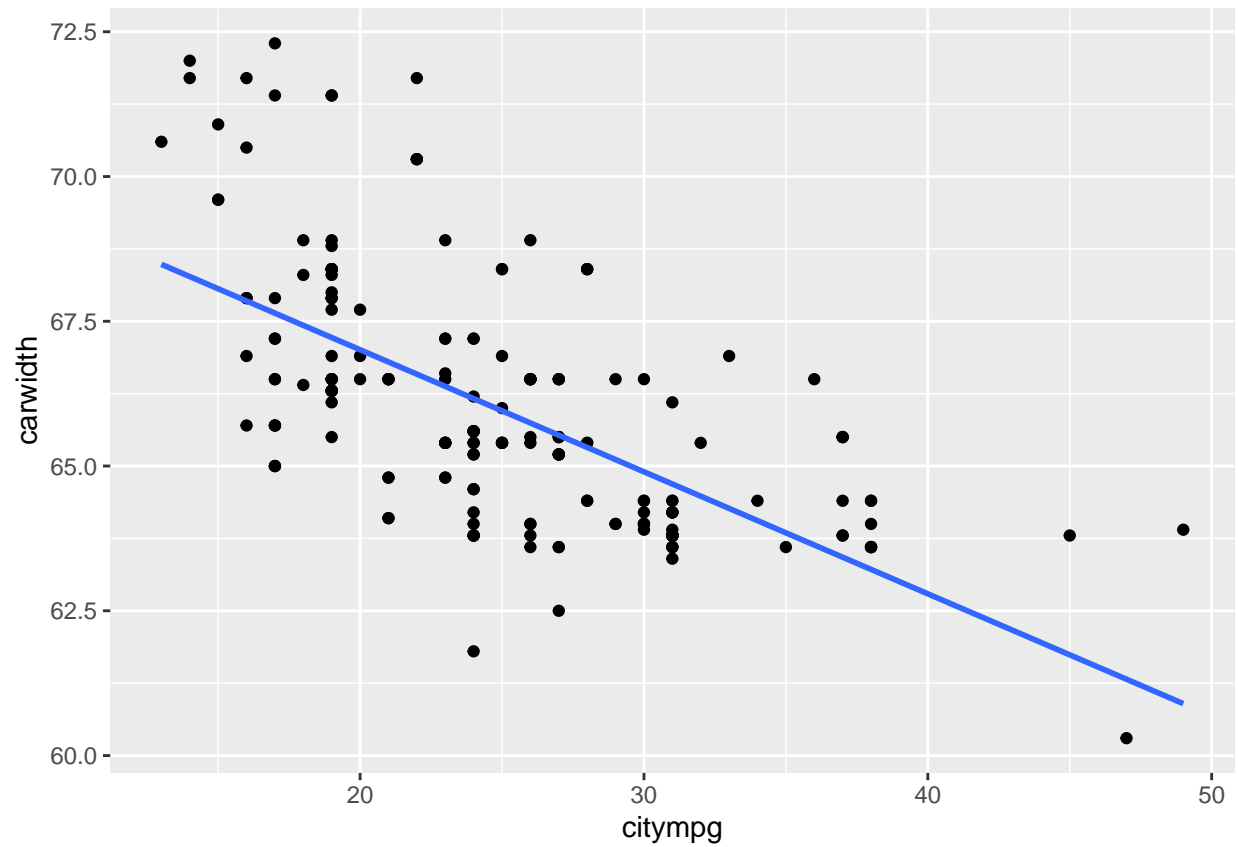
```
## [1] "Correlation between horsepower and curbweight : 0.750739251443481"  
## [1] "x = horsepower , y = enginesize"  
  
## 'geom_smooth()' using formula = 'y ~ x'
```



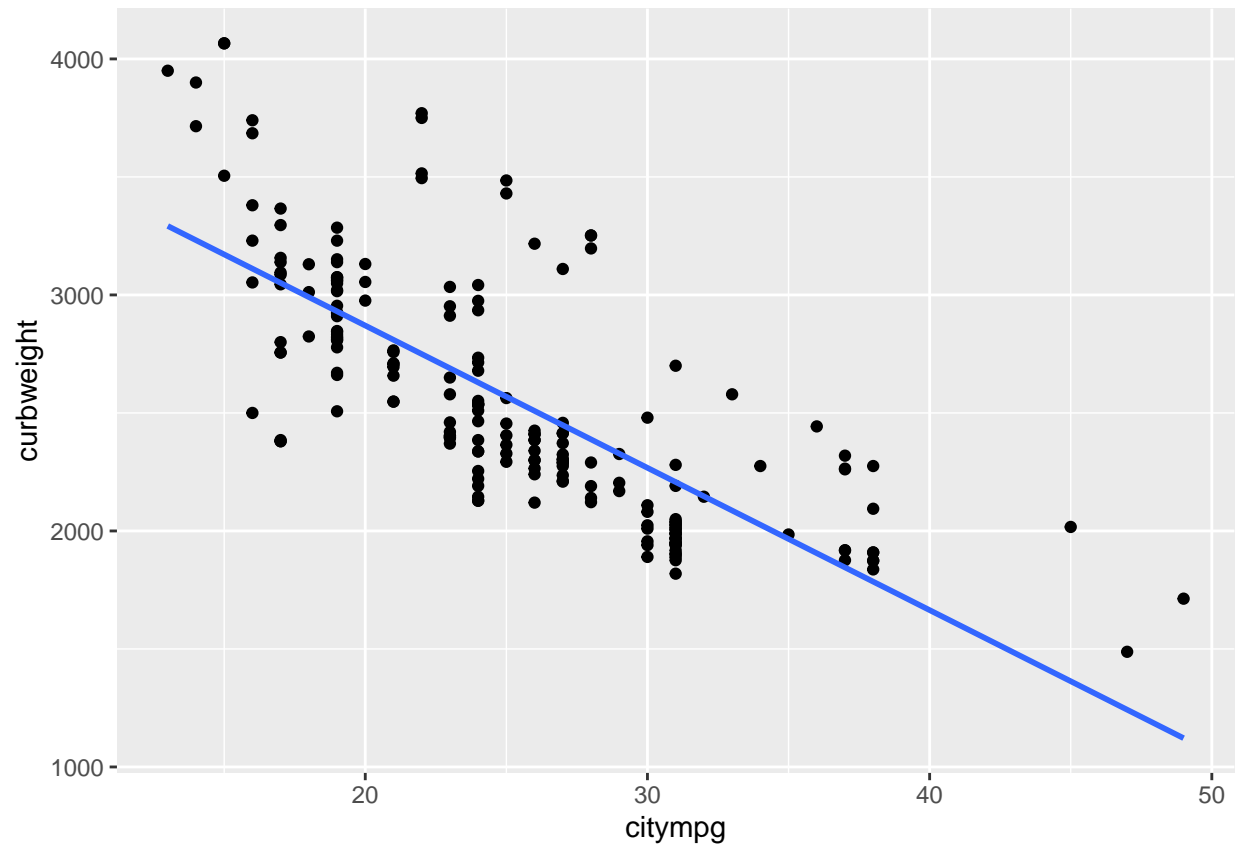
```
## [1] "Correlation between horsepower and enginesize : 0.80976865453773"  
## [1] "x = horsepower , y = citympg"  
  
## 'geom_smooth()' using formula = 'y ~ x'
```



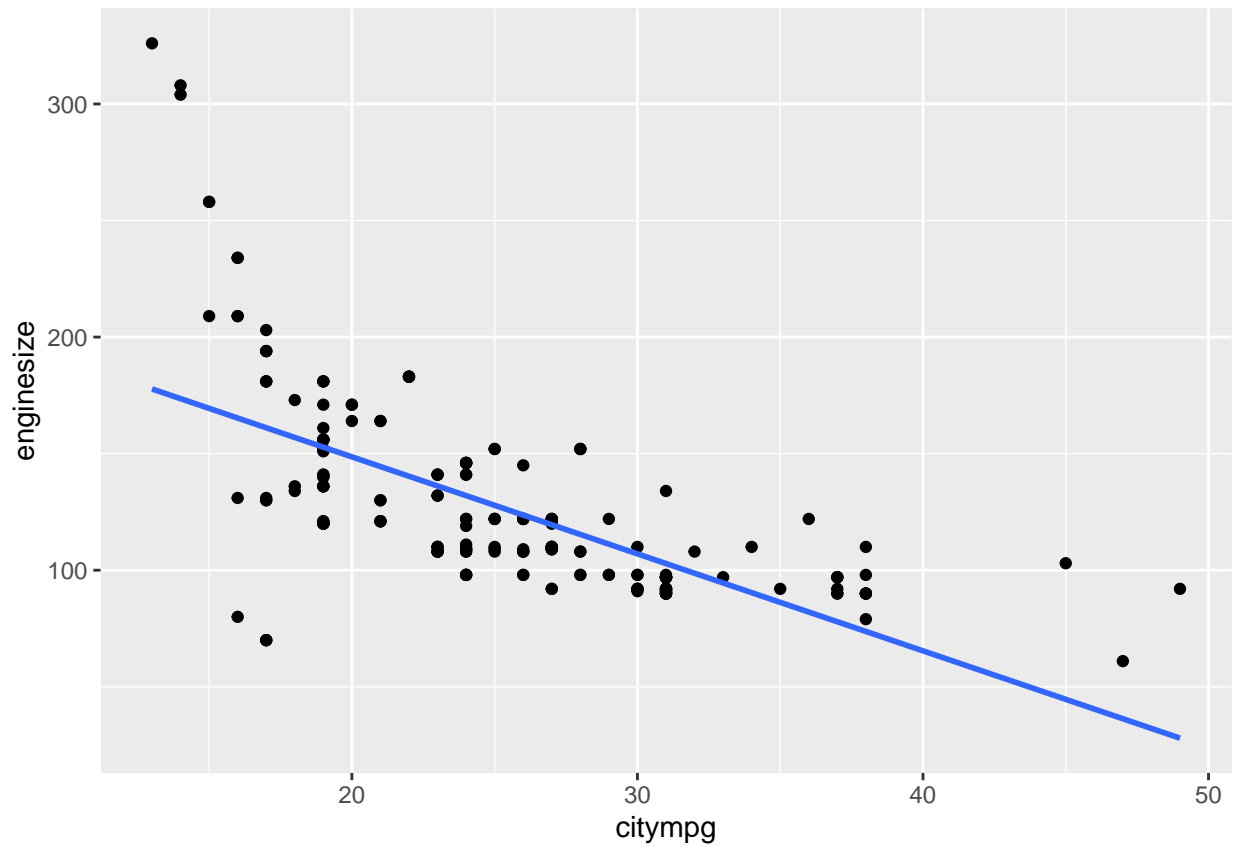
```
## [1] "Correlation between horsepower and citympg : -0.80145617566627"  
## [1] "x = citympg , y = carwidth"  
  
## 'geom_smooth()' using formula = 'y ~ x'
```



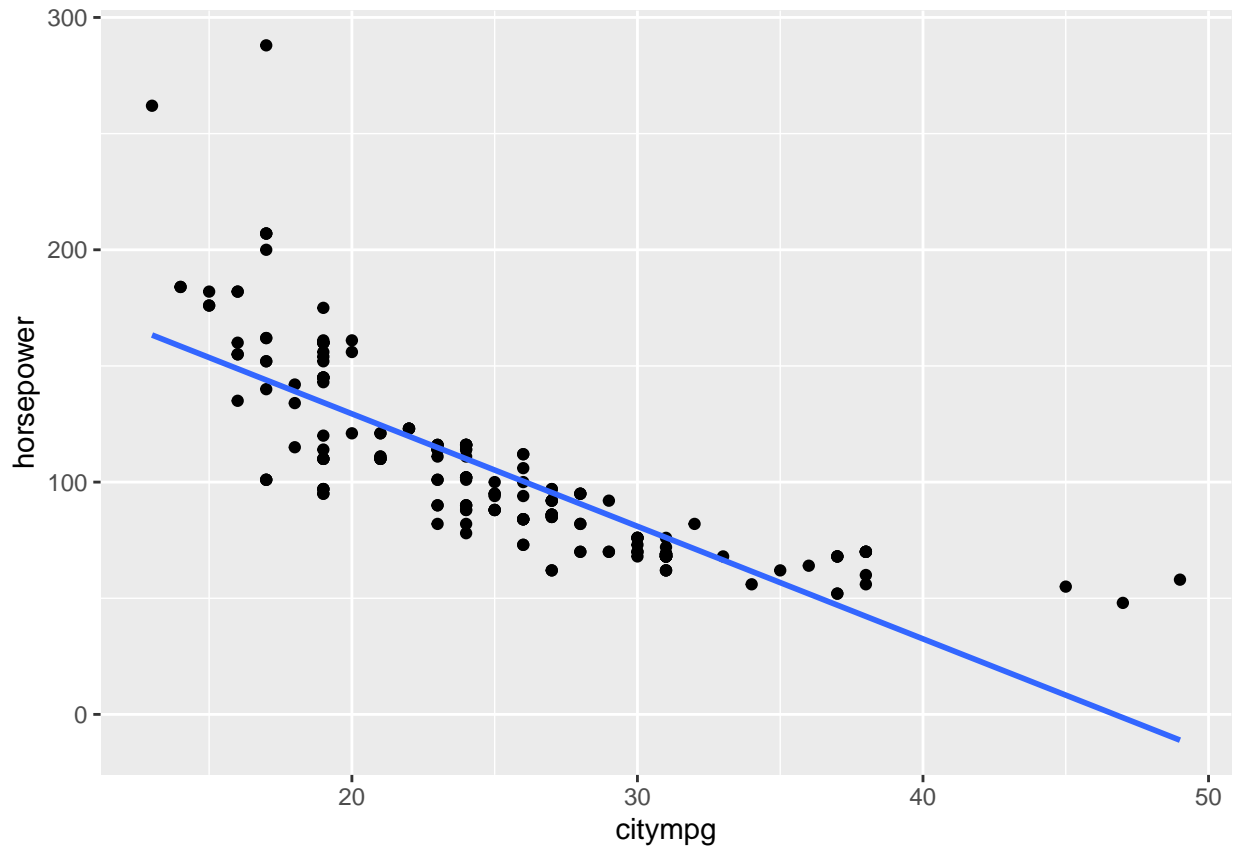
```
## [1] "Correlation between citympg and carwidth : -0.642704340710898"  
## [1] "x = citympg , y = curbweight"  
  
## 'geom_smooth()' using formula = 'y ~ x'
```

```
## [1] "Correlation between citympg and curbweight : -0.757413784505601"  
## [1] "x = citympg , y = enginesize"  
  
## 'geom_smooth()' using formula = 'y ~ x'
```



```
## [1] "Correlation between citympg and enginesize : -0.65365791631142"  
## [1] "x = citympg , y = horsepower"  
  
## 'geom_smooth()' using formula = 'y ~ x'
```



```
## [1] "Correlation between citympg and horsepower : -0.80145617566627"
```

cars93 - Correlation between pairs of variables except price

```
for (var in c("MPG.city", "EngineSize", "Horsepower", "Fuel.tank.capacity", "Weight")) {
  for (var1 in c("MPG.city", "EngineSize", "Horsepower", "Fuel.tank.capacity", "Weight")) {

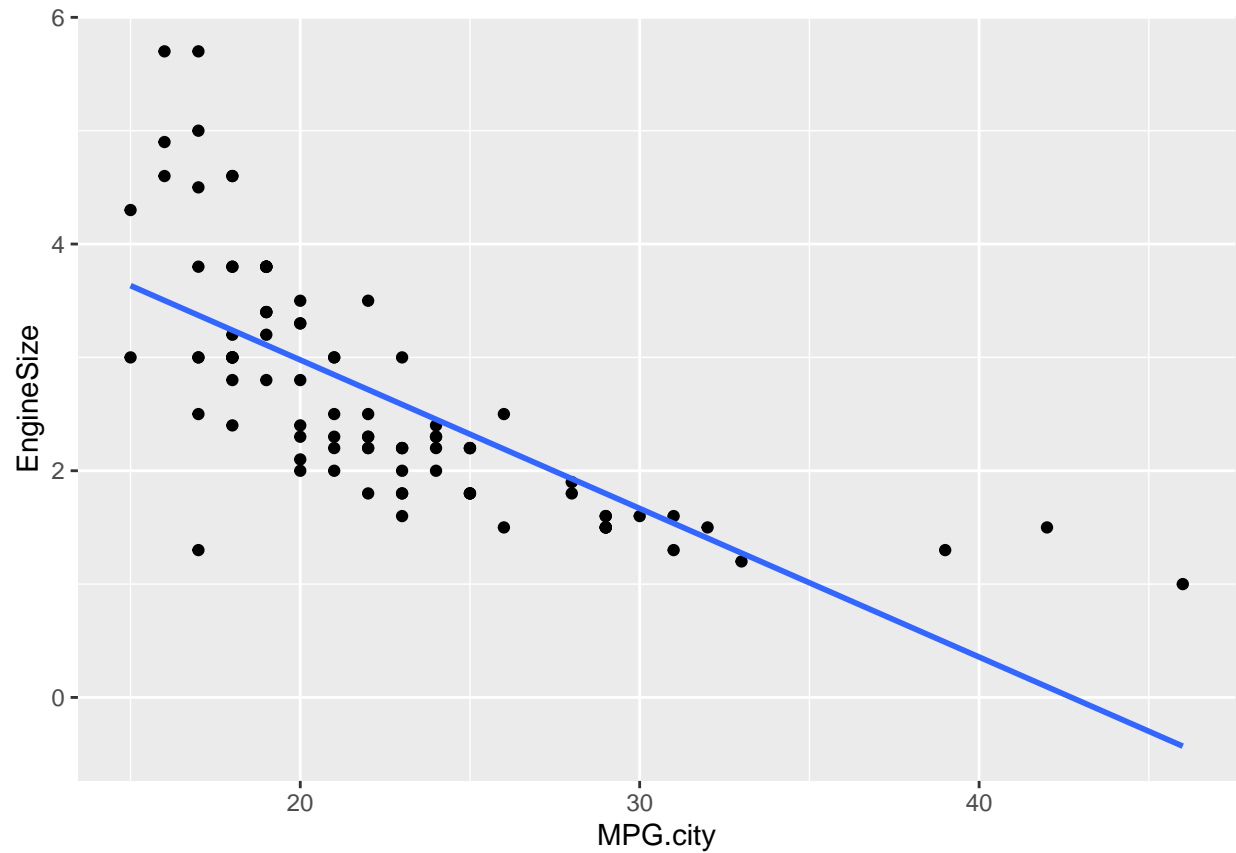
    if (var != var1) {
      print(paste("x =", var, ", y =", var1))

      p <- ggplot(cars93, aes_string(x = var, y = var1)) +
        geom_point() +
        geom_smooth(method = "lm", se = FALSE)
      print(p)

      correlation <- cor(cars93[[var1]], cars93[[var]])
      print(paste("Correlation between", var, "and", var1, ":", correlation))
    }
  }
}
```

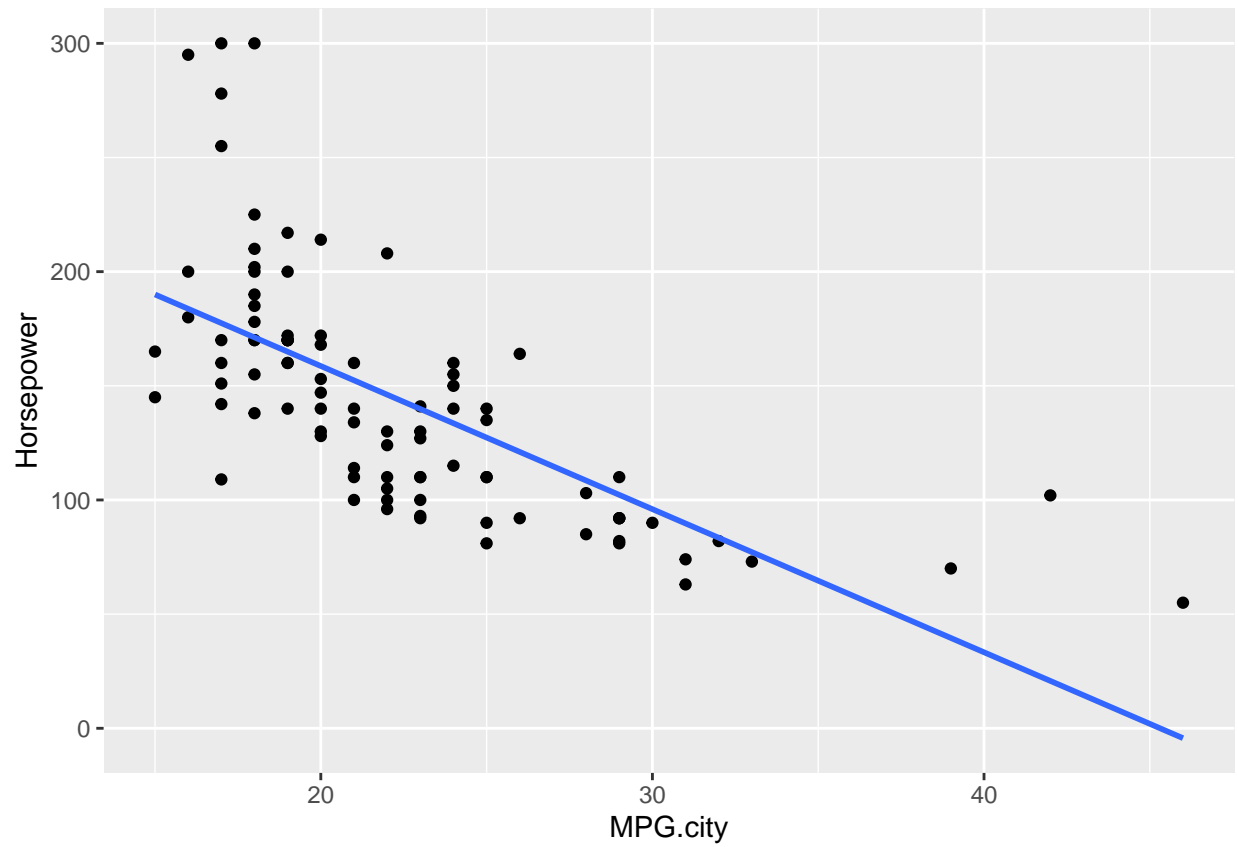
```
## [1] "x = MPG.city , y = EngineSize"
```

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

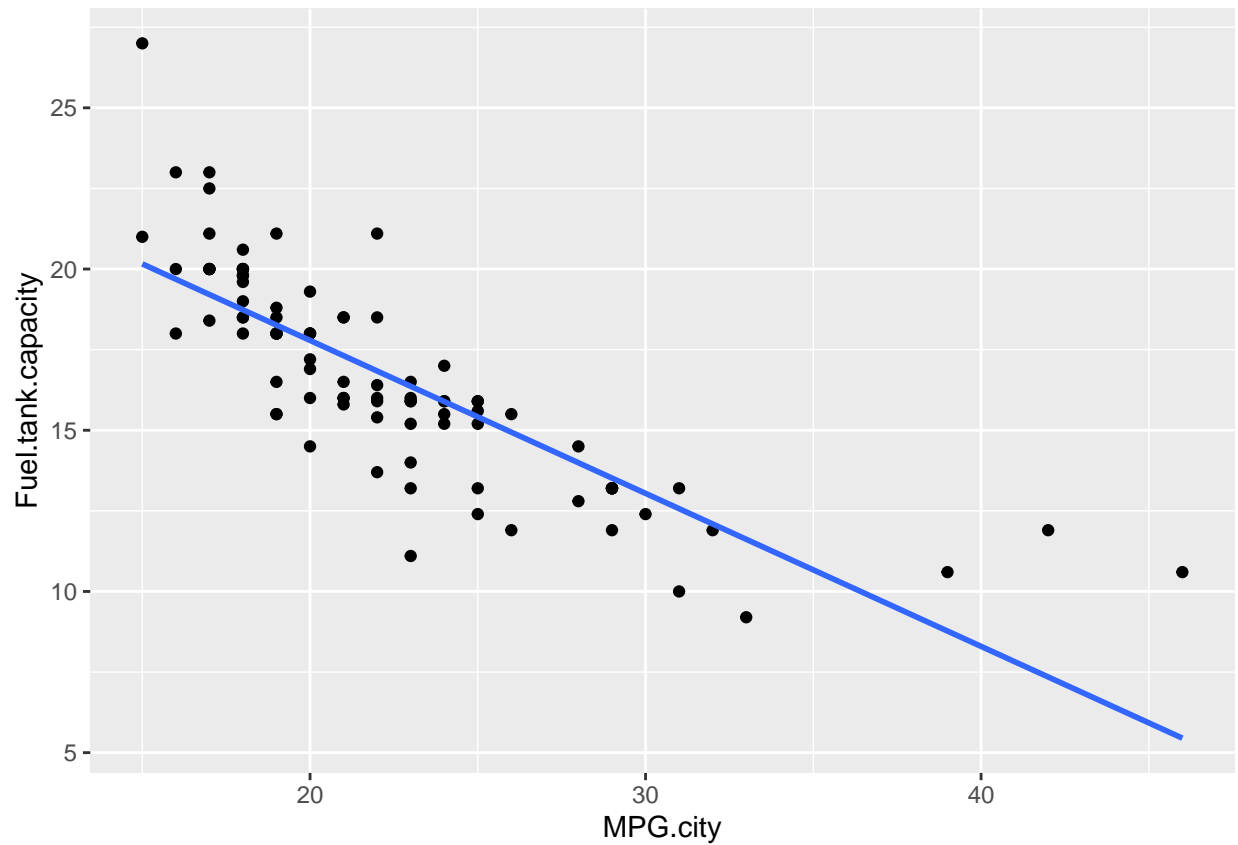


```
## [1] "Correlation between MPG.city and EngineSize : -0.710003161620369"  
## [1] "x = MPG.city , y = Horsepower"
```

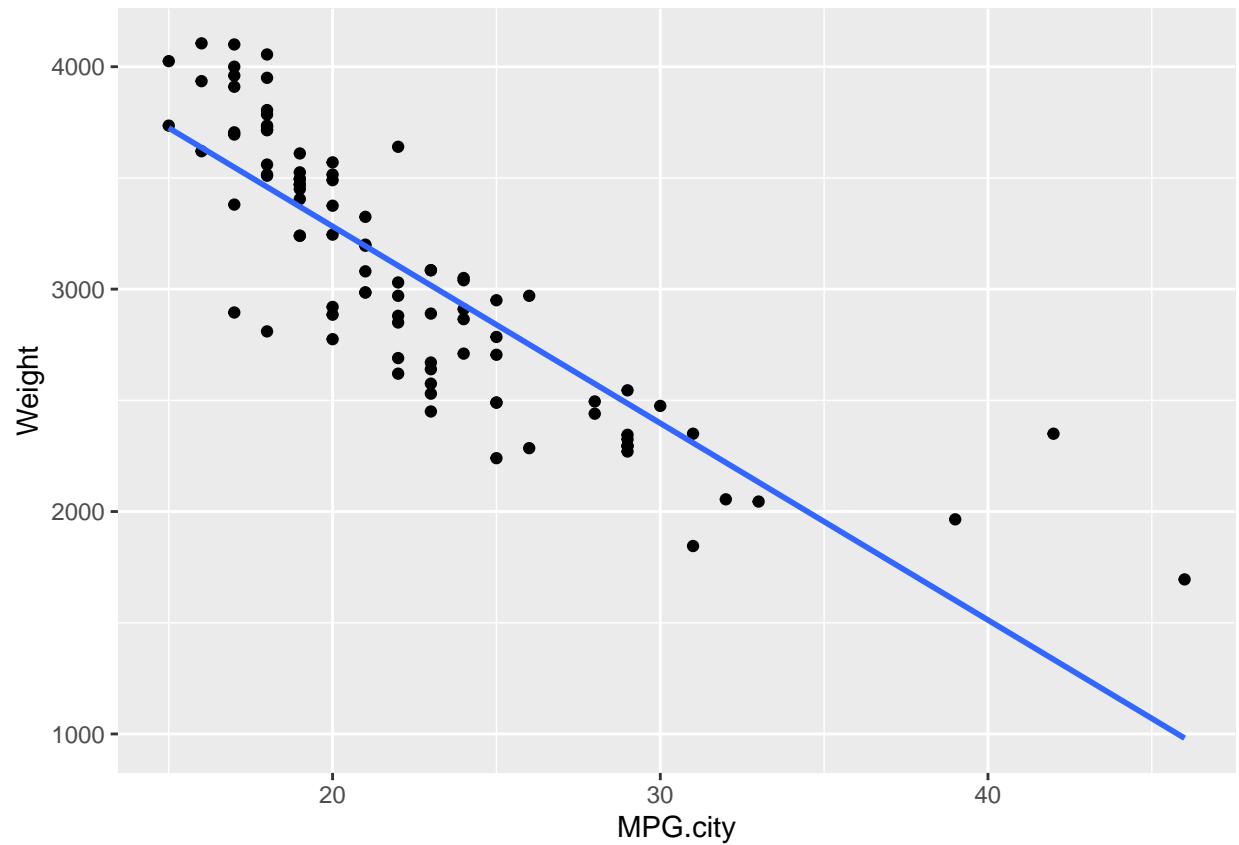
```
## 'geom_smooth()' using formula = 'y ~ x'
```



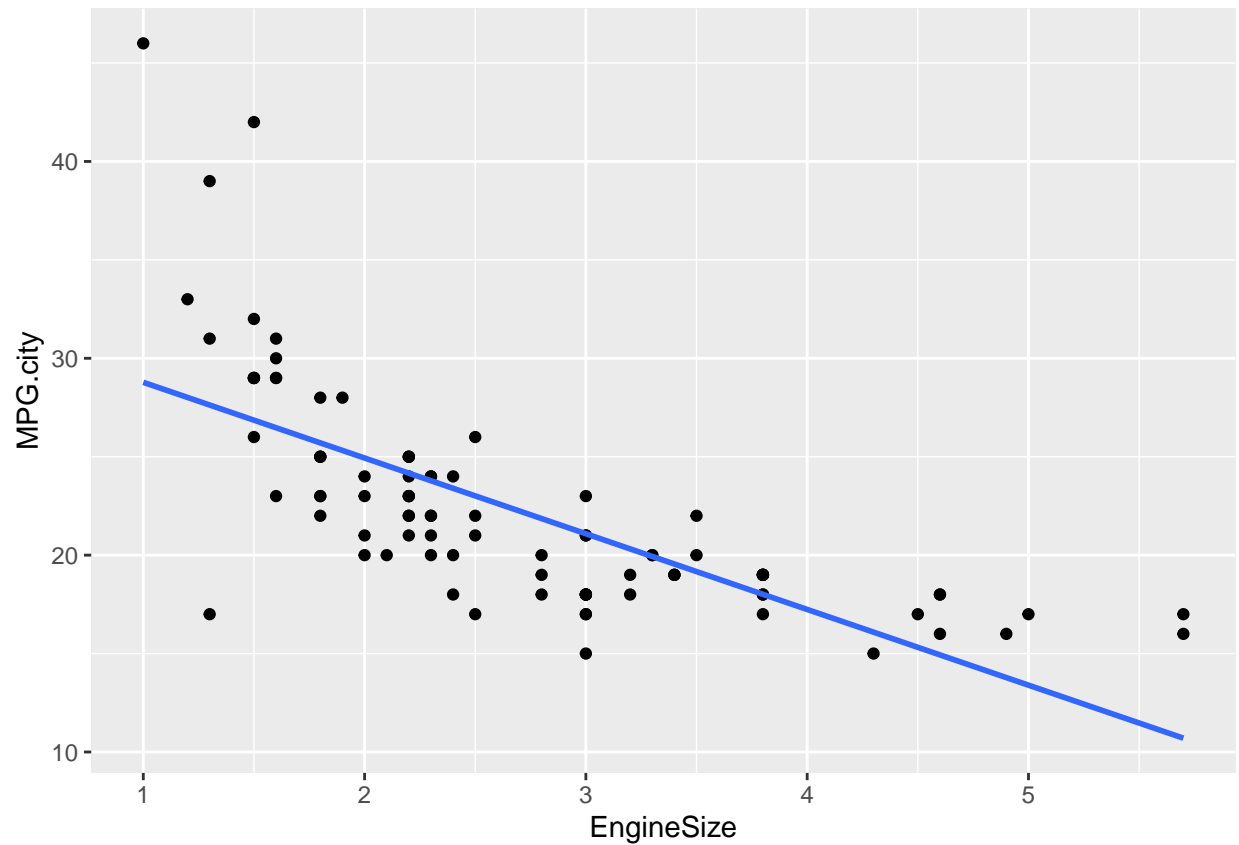
```
## [1] "Correlation between MPG.city and Horsepower : -0.672636150795724"  
## [1] "x = MPG.city , y = Fuel.tank.capacity"  
  
## 'geom_smooth()' using formula = 'y ~ x'
```



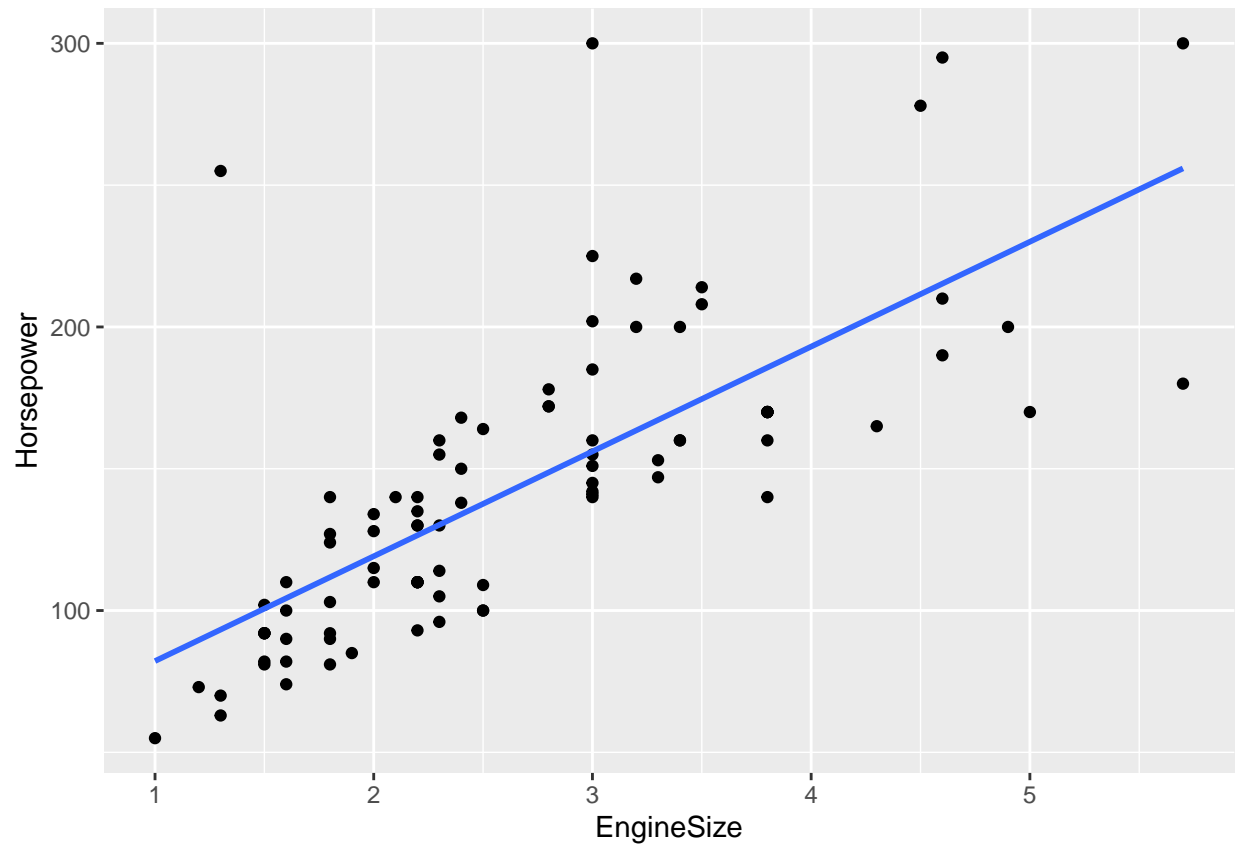
```
## [1] "Correlation between MPG.city and Fuel.tank.capacity : -0.813144416992242"  
## [1] "x = MPG.city , y = Weight"  
  
## 'geom_smooth()' using formula = 'y ~ x'
```



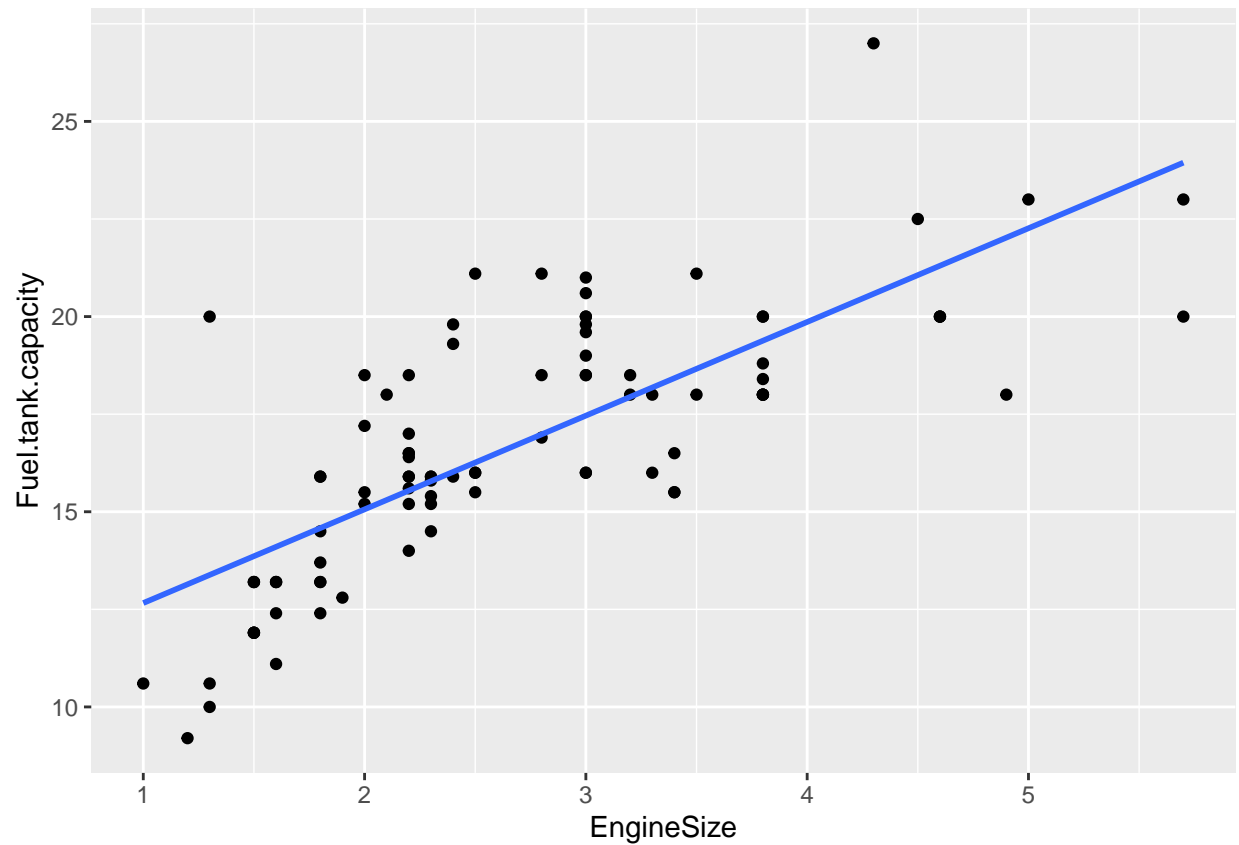
```
## [1] "Correlation between MPG.city and Weight : -0.84313854799682"  
## [1] "x = EngineSize , y = MPG.city"  
  
## 'geom_smooth()' using formula = 'y ~ x'
```



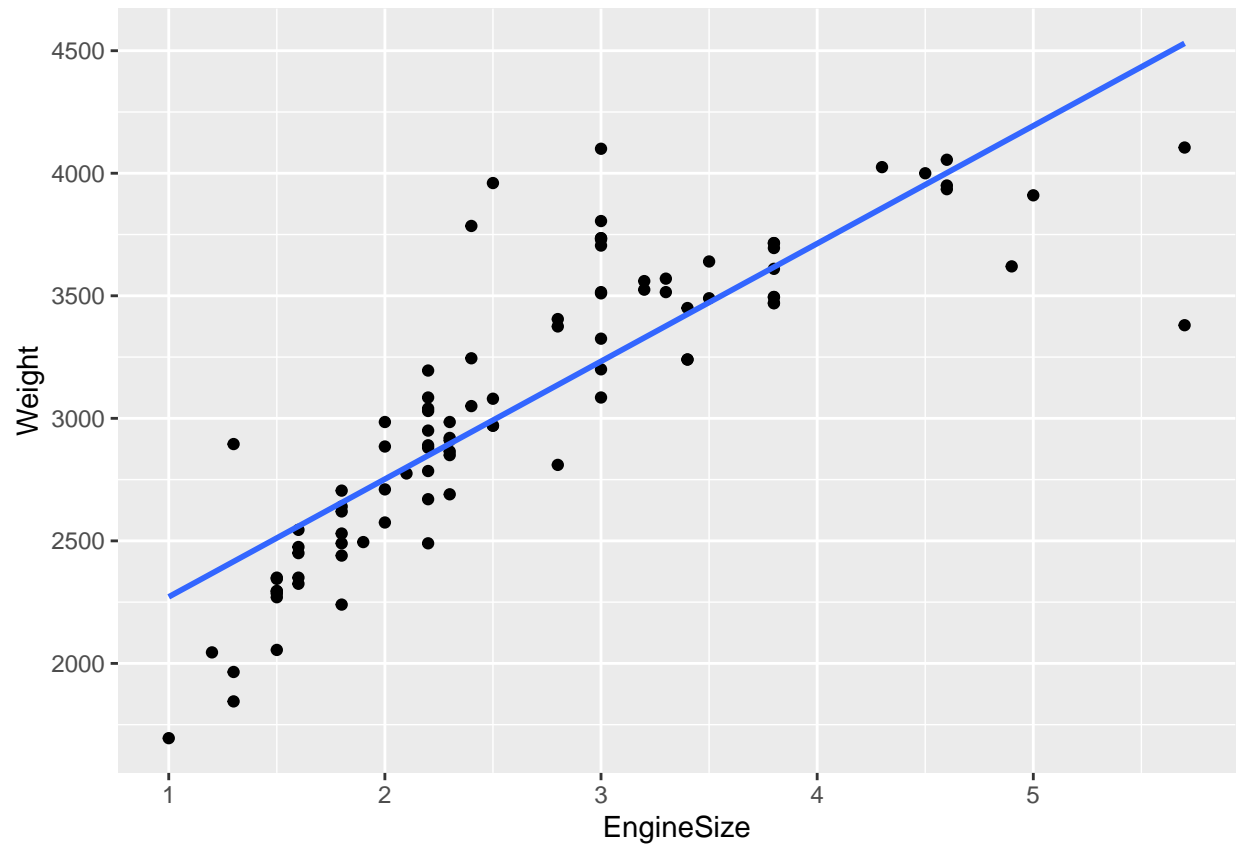
```
## [1] "Correlation between EngineSize and MPG.city : -0.710003161620369"  
## [1] "x = EngineSize , y = Horsepower"  
  
## 'geom_smooth()' using formula = 'y ~ x'
```

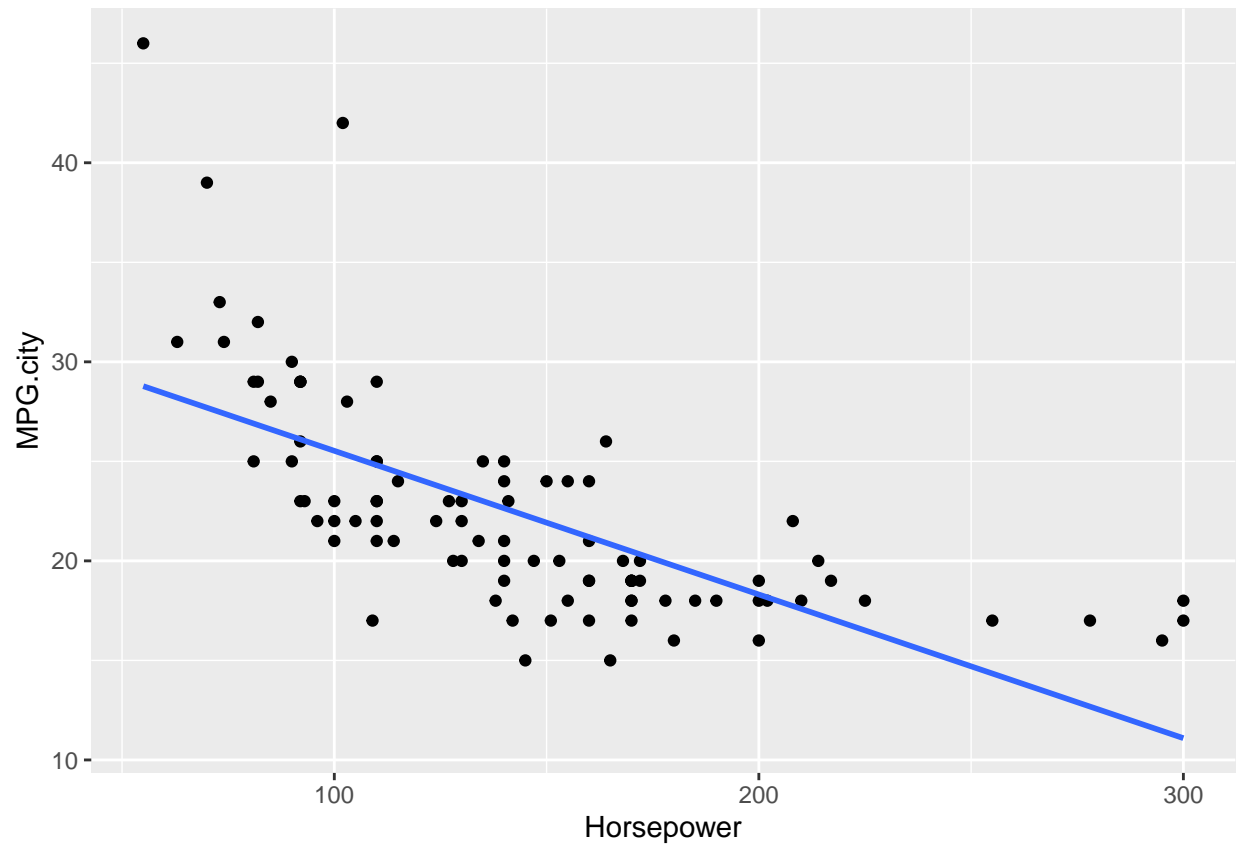
```
## [1] "Correlation between EngineSize and Horsepower : 0.732119730417387"  
## [1] "x = EngineSize , y = Fuel.tank.capacity"  
  
## 'geom_smooth()' using formula = 'y ~ x'
```



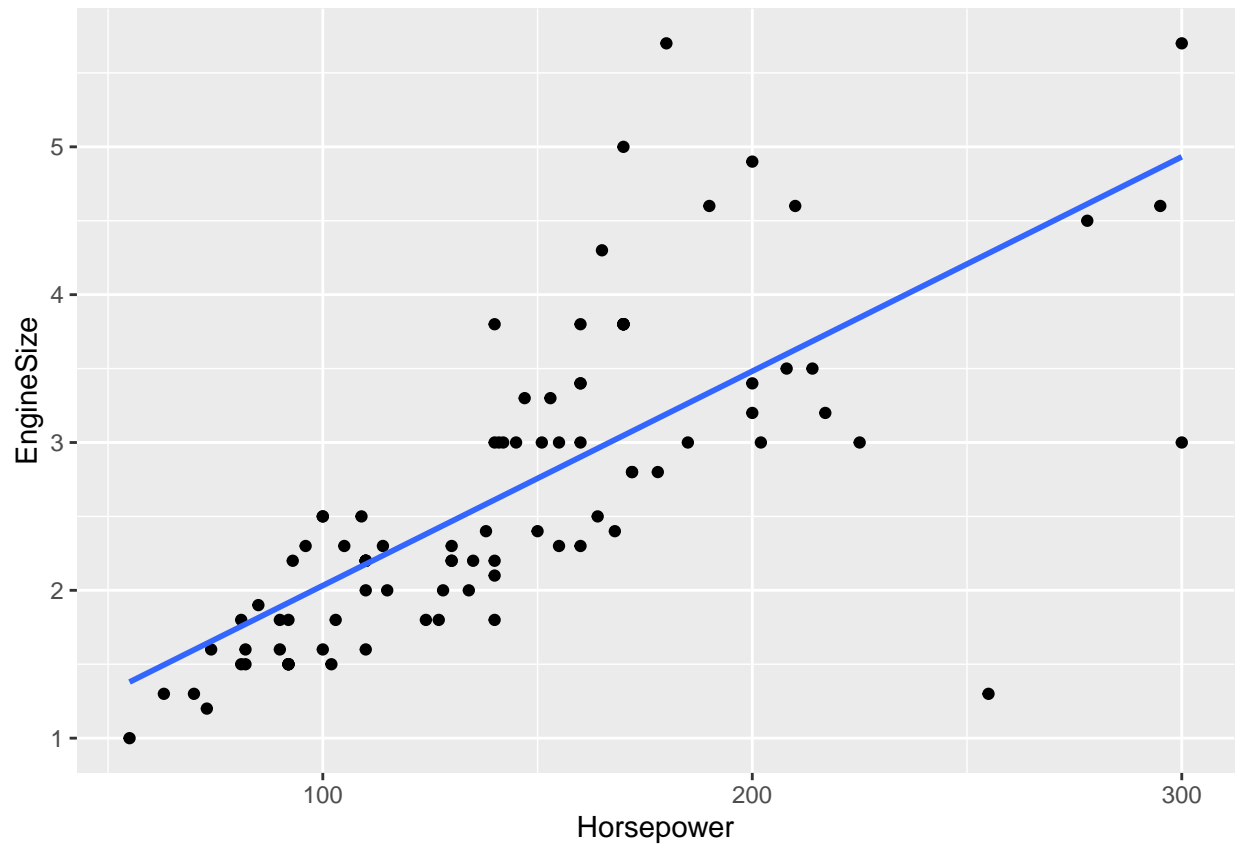
```
## [1] "Correlation between EngineSize and Fuel.tank.capacity : 0.759306224903401"  
## [1] "x = EngineSize , y = Weight"  
  
## 'geom_smooth()' using formula = 'y ~ x'
```



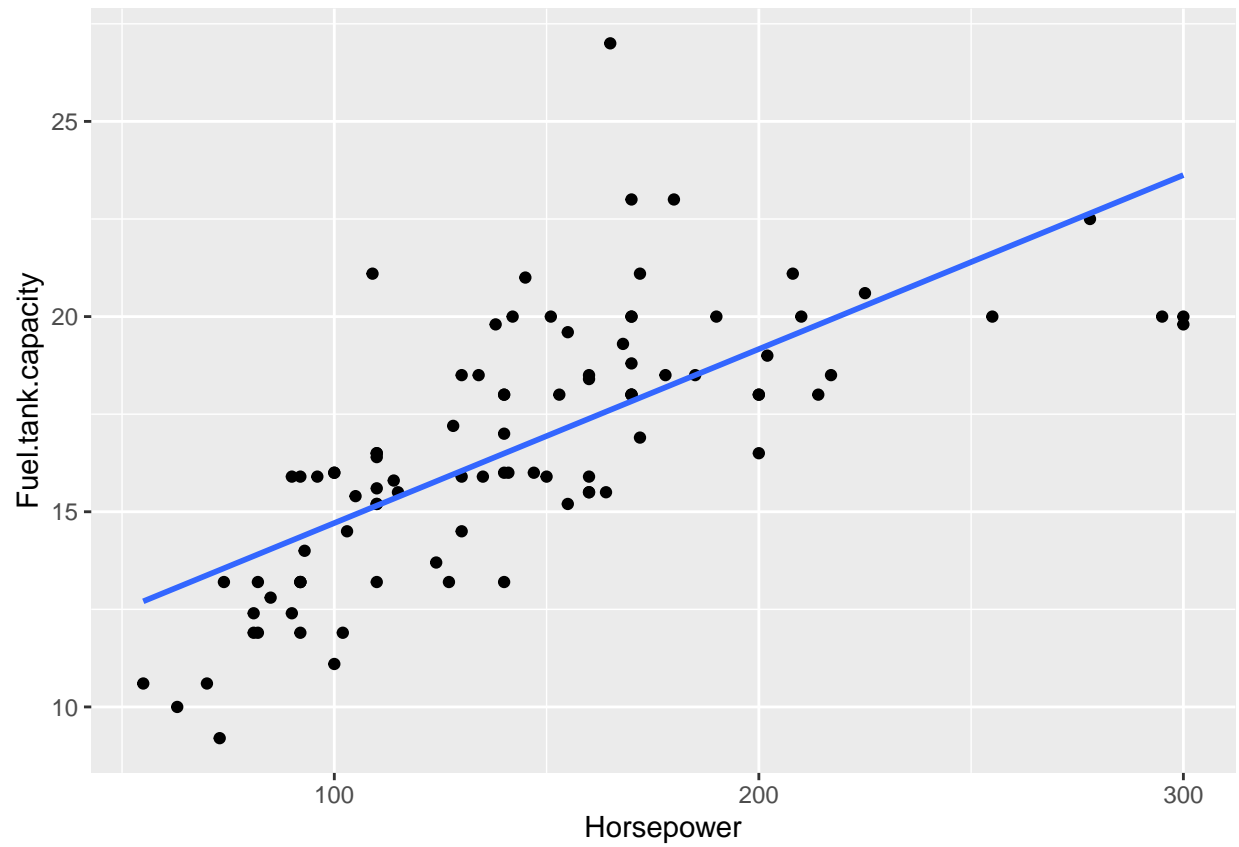
```
## [1] "Correlation between EngineSize and Weight : 0.845075335037262"  
## [1] "x = Horsepower , y = MPG.city"  
  
## 'geom_smooth()' using formula = 'y ~ x'
```



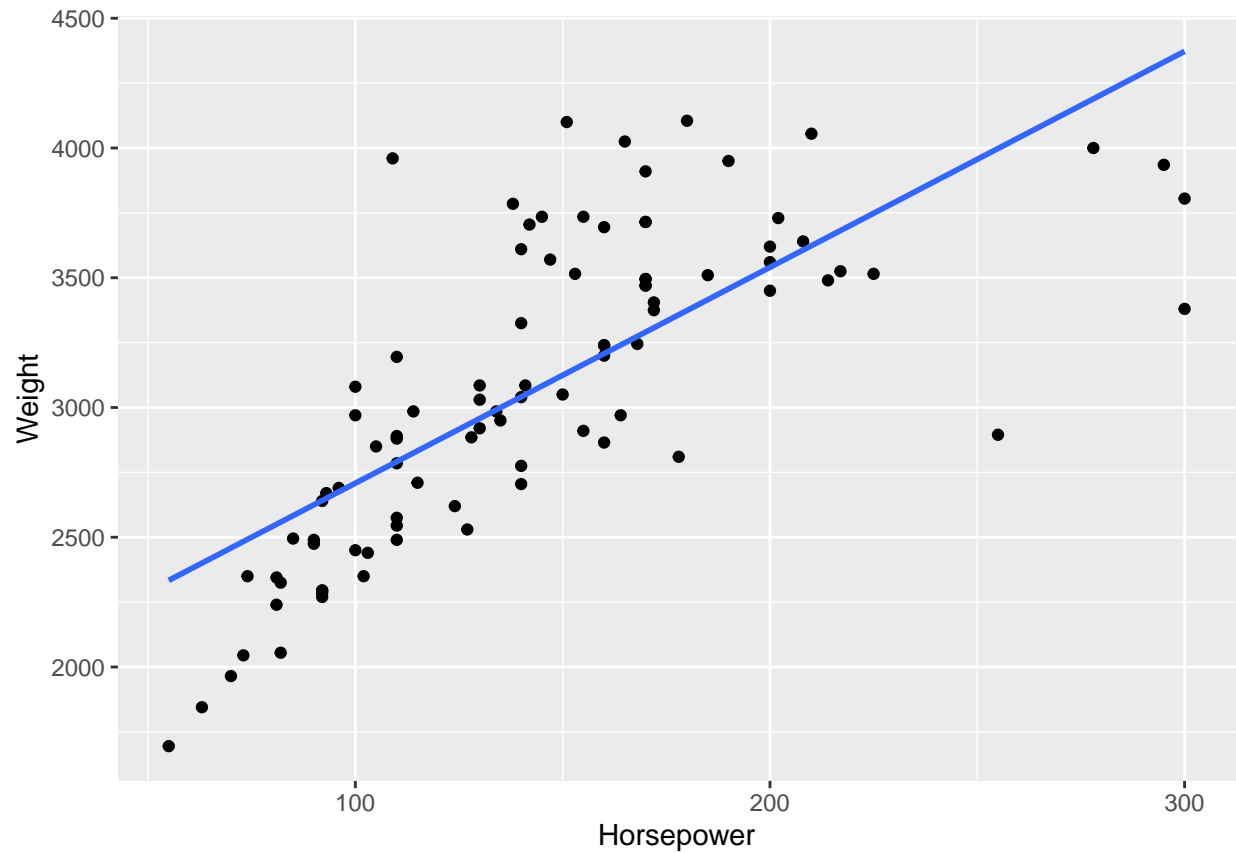
```
## [1] "Correlation between Horsepower and MPG.city : -0.672636150795724"  
## [1] "x = Horsepower , y = EngineSize"  
  
## 'geom_smooth()' using formula = 'y ~ x'
```



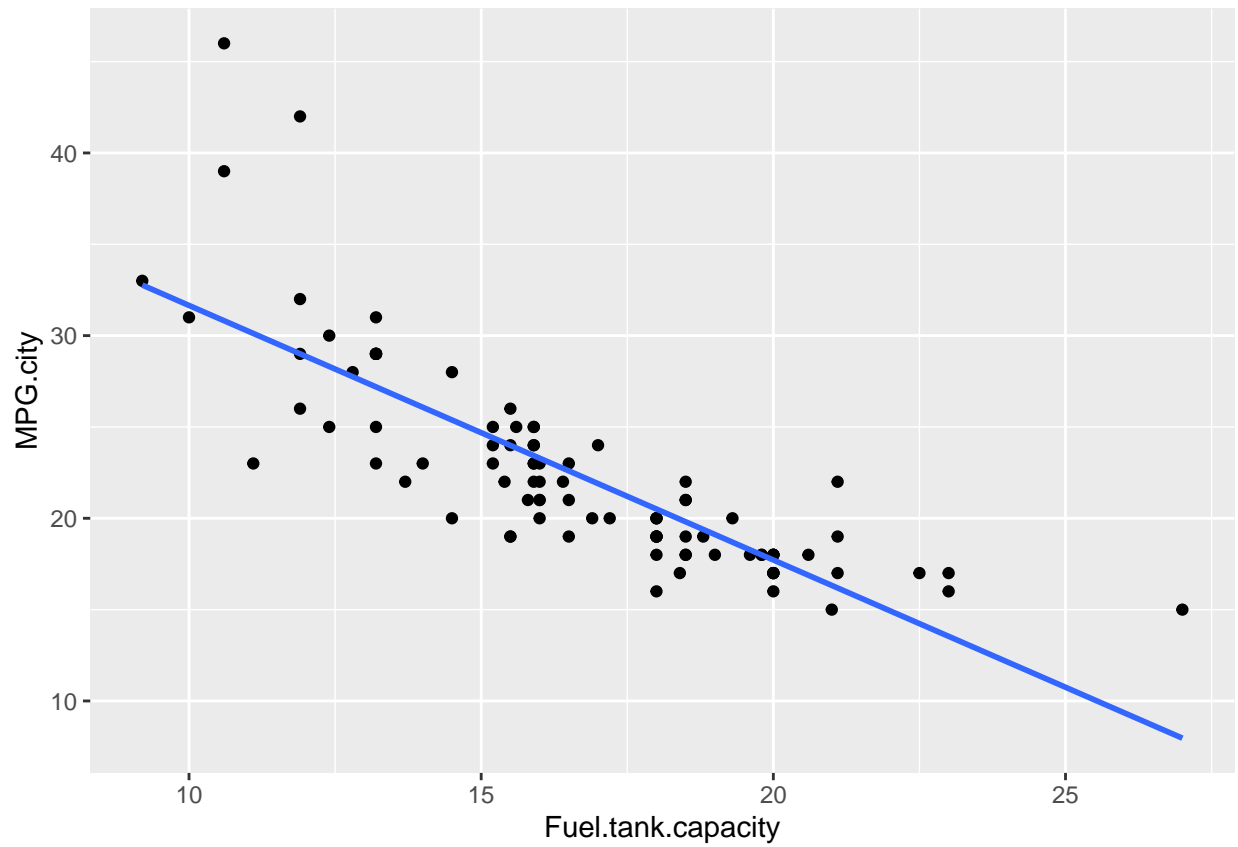
```
## [1] "Correlation between Horsepower and EngineSize : 0.732119730417387"  
## [1] "x = Horsepower , y = Fuel.tank.capacity"  
  
## 'geom_smooth()' using formula = 'y ~ x'
```



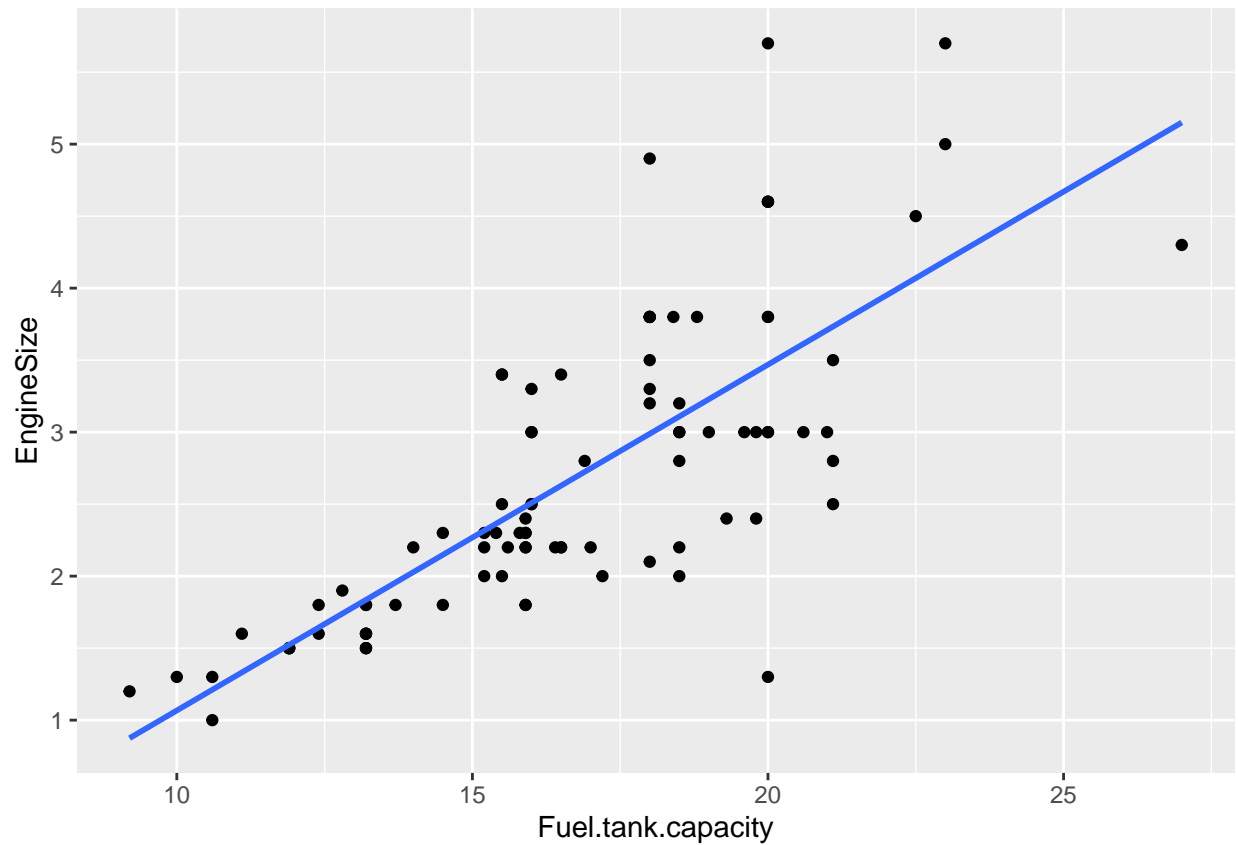
```
## [1] "Correlation between Horsepower and Fuel.tank.capacity : 0.711790317259719"  
## [1] "x = Horsepower , y = Weight"  
  
## 'geom_smooth()' using formula = 'y ~ x'
```



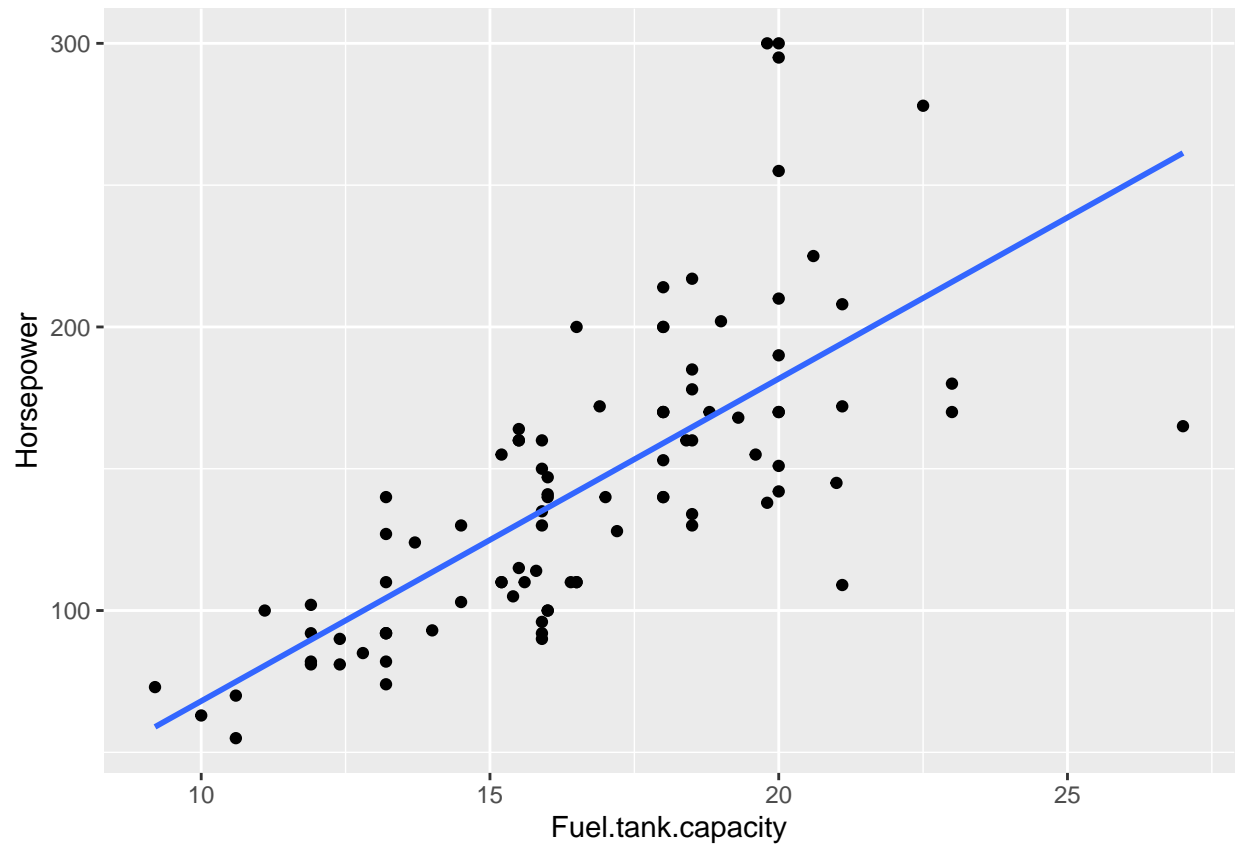
```
## [1] "Correlation between Horsepower and Weight : 0.738797515540372"  
## [1] "x = Fuel.tank.capacity , y = MPG.city"  
  
## 'geom_smooth()' using formula = 'y ~ x'
```



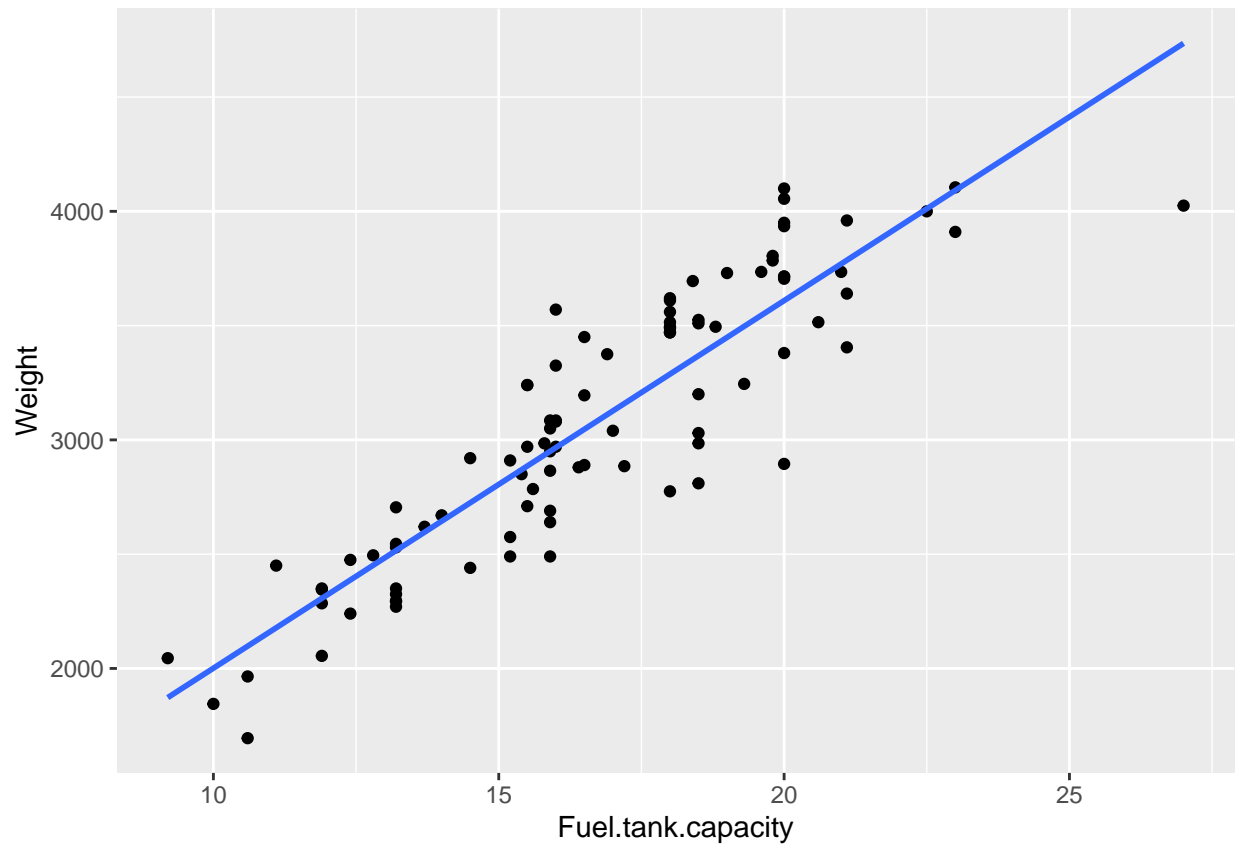
```
## [1] "Correlation between Fuel.tank.capacity and MPG.city : -0.813144416992242"  
## [1] "x = Fuel.tank.capacity , y = EngineSize"  
  
## 'geom_smooth()' using formula = 'y ~ x'
```

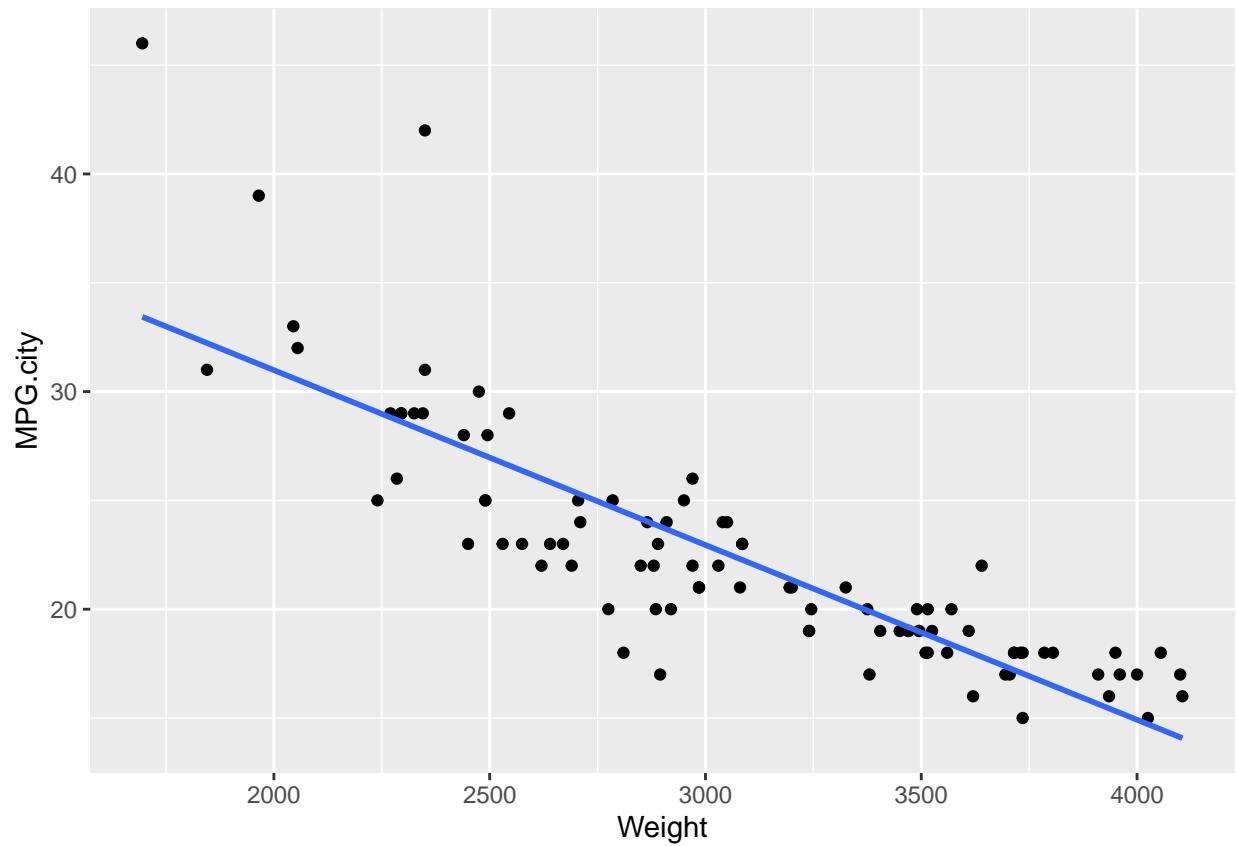
```
## [1] "Correlation between Fuel.tank.capacity and EngineSize : 0.759306224903401"  
## [1] "x = Fuel.tank.capacity , y = Horsepower"  
  
## 'geom_smooth()' using formula = 'y ~ x'
```



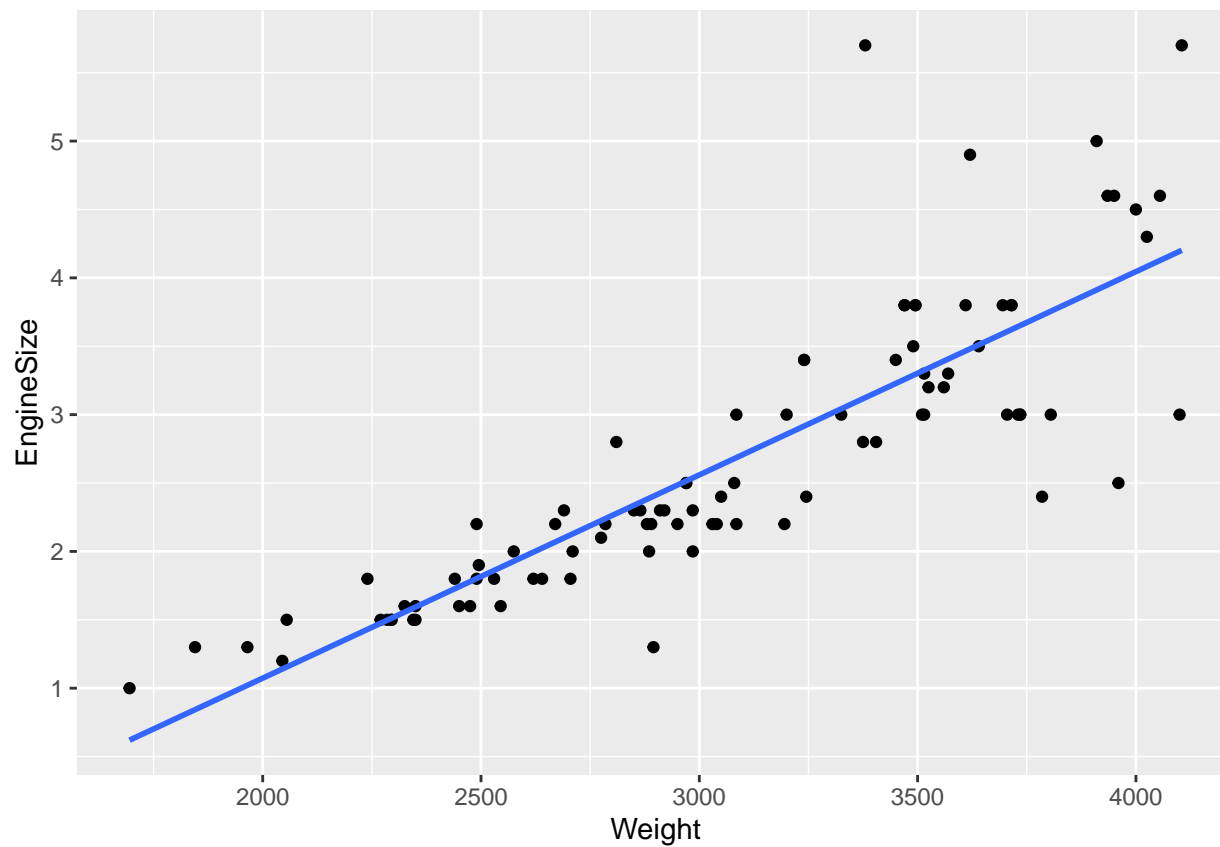
```
## [1] "Correlation between Fuel.tank.capacity and Horsepower : 0.711790317259719"  
## [1] "x = Fuel.tank.capacity , y = Weight"  
  
## 'geom_smooth()' using formula = 'y ~ x'
```



```
## [1] "Correlation between Fuel.tank.capacity and Weight : 0.894018054491314"  
## [1] "x = Weight , y = MPG.city"  
  
## 'geom_smooth()' using formula = 'y ~ x'
```

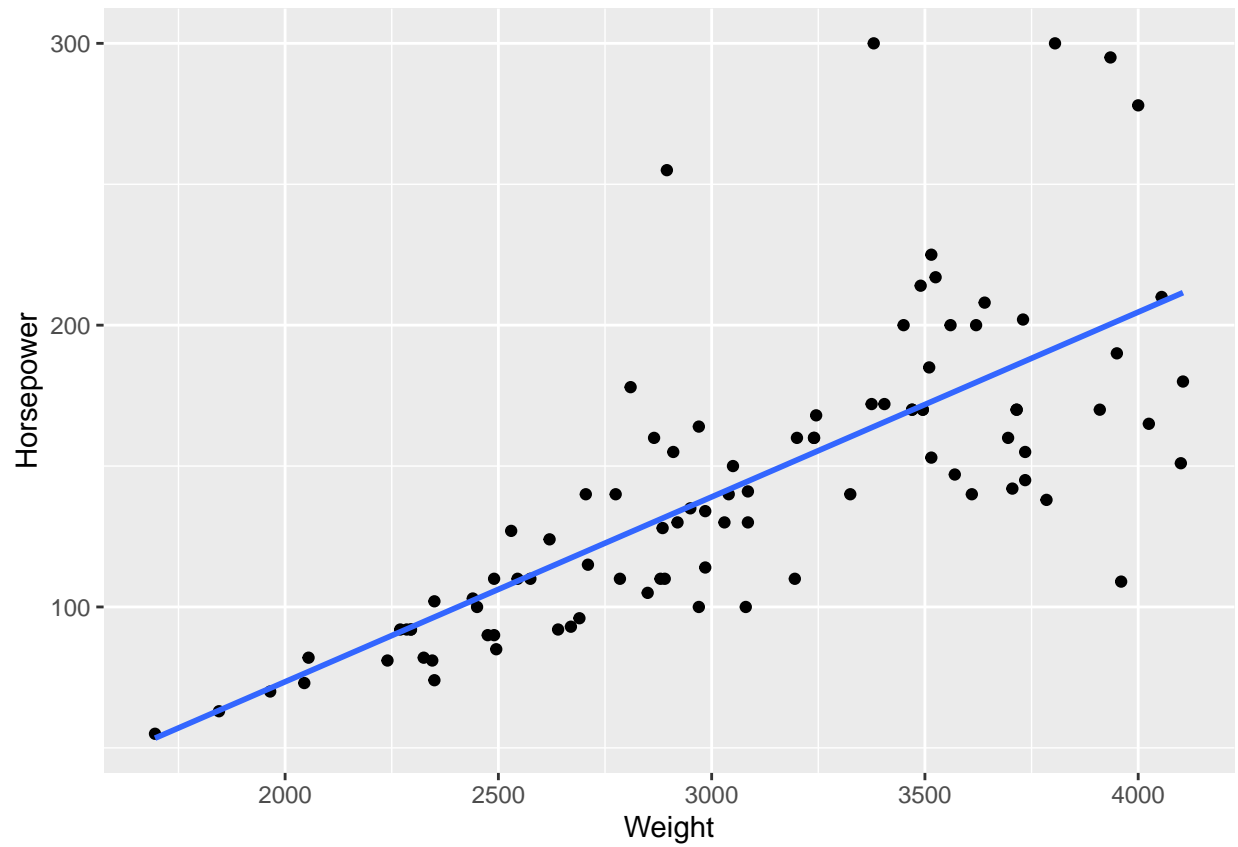


```
## [1] "Correlation between Weight and MPG.city : -0.84313854799682"  
## [1] "x = Weight , y = EngineSize"  
  
## 'geom_smooth()' using formula = 'y ~ x'
```

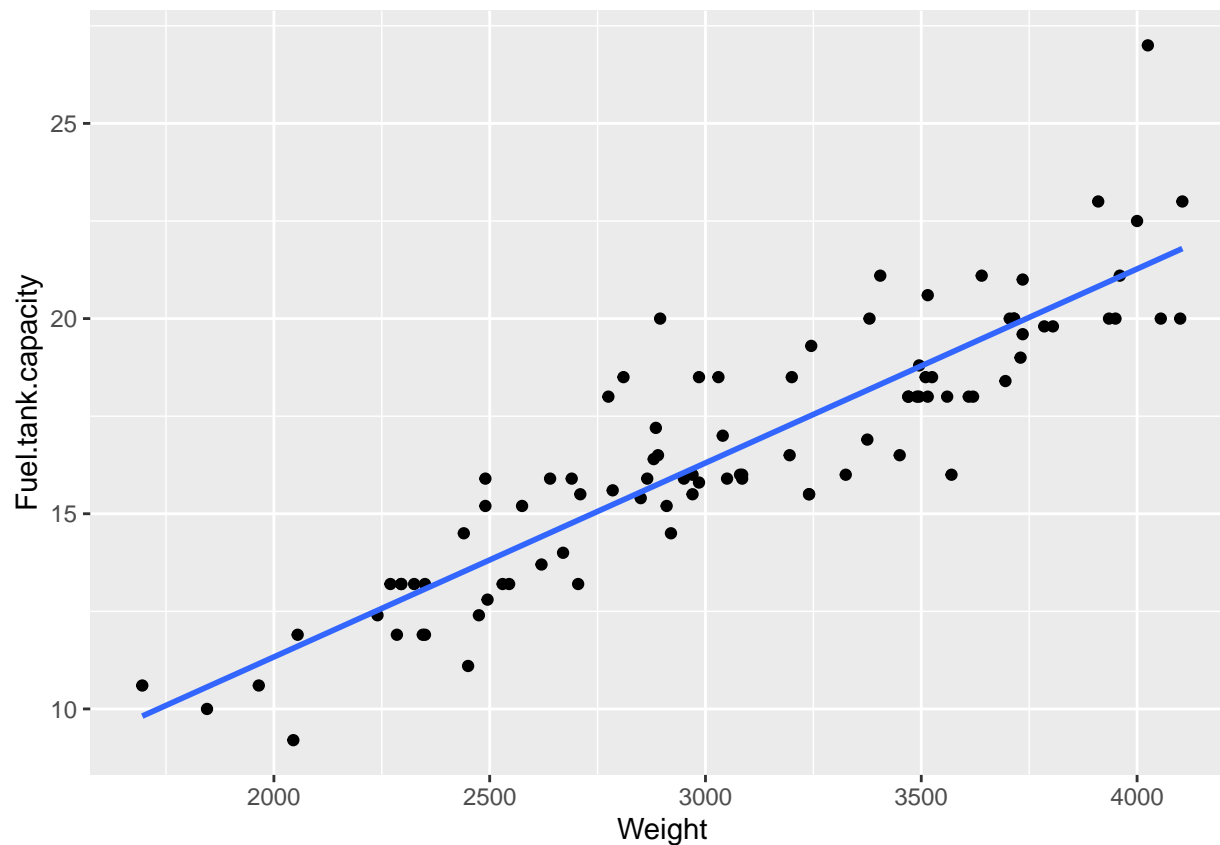


```
## [1] "Correlation between Weight and EngineSize : 0.845075335037262"  
## [1] "x = Weight , y = Horsepower"
```

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



```
## [1] "Correlation between Weight and Horsepower : 0.738797515540372"  
## [1] "x = Weight , y = Fuel.tank.capacity"  
  
## 'geom_smooth()' using formula = 'y ~ x'
```



```
## [1] "Correlation between Weight and Fuel.tank.capacity : 0.894018054491314"
```

Linear regression model using common columns: mpgcity+highway, horsepower, enginesize, weight

```
model_assign2 <- lm(price ~ citympg + highwaympg + curbweight + enginesize + horsepower, data = cars_assign2)
summary(model_assign2)
```

```
##
## Call:
## lm(formula = price ~ citympg + highwaympg + curbweight + enginesize +
##     horsepower, data = cars_assign2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -8942.7 -1685.5   -35.5   1322.6 13679.1
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -15803.388   4053.921  -3.898 0.000132 ***
## citympg      -40.841    177.895  -0.230 0.818657
## highwaympg    75.944    167.738   0.453 0.651218
## curbweight     4.654     1.101   4.226 3.63e-05 ***
## enginesize    83.295    13.586   6.131 4.61e-09 ***
## horsepower    51.005    13.878   3.675 0.000305 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3486 on 199 degrees of freedom
## Multiple R-squared:  0.8142, Adjusted R-squared:  0.8095
## F-statistic: 174.4 on 5 and 199 DF,  p-value: < 2.2e-16

summary(model_assign2)

##
## Call:
## lm(formula = price ~ citympg + highwaympg + curbweight + enginesize +
##     horsepower, data = cars_assign)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -8942.7 -1685.5   -35.5   1322.6 13679.1
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -15803.388   4053.921  -3.898 0.000132 ***
## citympg      -40.841    177.895   -0.230 0.818657
## highwaympg    75.944    167.738    0.453 0.651218
## curbweight     4.654     1.101    4.226 3.63e-05 ***
## enginesize    83.295     13.586    6.131 4.61e-09 ***
## horsepower    51.005     13.878    3.675 0.000305 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3486 on 199 degrees of freedom
## Multiple R-squared:  0.8142, Adjusted R-squared:  0.8095
## F-statistic: 174.4 on 5 and 199 DF,  p-value: < 2.2e-16
```

The main model

```
model_assign3 <- lm(price ~ curbweight + enginesize + horsepower, data = cars_assign)
summary(model_assign3)
```

```
##
## Call:
## lm(formula = price ~ curbweight + enginesize + horsepower, data = cars_assign)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
##  -9003   -1701    -24    1340   13760
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.346e+04  1.333e+03 -10.100 < 2e-16 ***
## curbweight   4.263e+00  9.065e-01  4.702 4.78e-06 ***
## enginesize   8.488e+01  1.276e+01  6.651 2.69e-10 ***
## horsepower   4.875e+01  1.070e+01  4.557 8.99e-06 ***
```



```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3473 on 201 degrees of freedom
## Multiple R-squared:  0.8138, Adjusted R-squared:  0.811
## F-statistic: 292.9 on 3 and 201 DF,  p-value: < 2.2e-16

model_assign_all <- lm(price ~ wheelbase + carlength + carwidth + carheight + curbweight + enginesize +
summary(model_assign_all)

##
## Call:
## lm(formula = price ~ wheelbase + carlength + carwidth + carheight +
##     curbweight + enginesize + horsepower + peakrpm + citympg +
##     highwaympg, data = cars_assign)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -8806.6 -1643.9  -146.4   1355.1  13900.7
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6.787e+04  1.486e+04  -4.566 8.82e-06 ***
## wheelbase    4.487e+01  1.046e+02   0.429  0.66844
## carlength   -8.311e+01  5.812e+01  -1.430  0.15433
## carwidth     6.069e+02  2.567e+02   2.365  0.01904 *
## carheight    2.369e+02  1.413e+02   1.677  0.09524 .
## curbweight   3.916e+00  1.670e+00   2.345  0.02004 *
## enginesize    1.009e+02  1.413e+01   7.142 1.80e-11 ***
## horsepower   3.305e+01  1.658e+01   1.994  0.04756 *
## peakrpm      2.020e+00  6.713e-01   3.008  0.00297 **
## citympg     -1.202e+02  1.813e+02  -0.663  0.50822
## highwaympg   1.585e+02  1.672e+02   0.948  0.34449
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3364 on 194 degrees of freedom
## Multiple R-squared:  0.8314, Adjusted R-squared:  0.8227
## F-statistic: 95.68 on 10 and 194 DF,  p-value: < 2.2e-16

any(is.na(cars93))

## [1] TRUE

custom_model <- lm(price ~ enginesize, data = cars_assign)
summary(custom_model)

##
## Call:
## lm(formula = price ~ enginesize, data = cars_assign)
##
## Residuals:
```

```
##      Min      1Q   Median      3Q      Max
## -10664.2 -2225.0   -482.4   1588.0  14271.5
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -8005.446     873.221  -9.168  <2e-16 ***
## enginesize    167.698       6.539   25.645  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3889 on 203 degrees of freedom
## Multiple R-squared:  0.7641, Adjusted R-squared:  0.763
## F-statistic: 657.6 on 1 and 203 DF,  p-value: < 2.2e-16
```

Data Cleaning/n Clean Cars93.csv, use it to validate model.

```
price_prediction_model <- model_assign3
```

Get the columns needed to validate from cars 93, three needed columns

```
car_variables <- cars93[, c("Weight", "EngineSize", "Horsepower")]
```

Convert EngineSize to cubic centiliters

```
car_variables$EngineSize <- car_variables$EngineSize * 100
```

Change the column names to reflect that of cars__assign

```
colnames(car_variables) <- c("curbweight", "enginesize", "horsepower")
```

Check for null rows

```
anyNA(car_variables)
```

```
## [1] FALSE
```

Check cars 93 for NA

```
anyNA(cars93)
```

```
## [1] TRUE
```

Predict for first row

```
first_row = car_variables[1, ]
print(first_row)
```

```
##   curbweight enginesize horsepower
## 1       2705       180         140
```

```
first_row <- data.frame(curbweight = 2705, enginesize = 180, horsepower = 140)
```

```
predicted_price <- predict(price_prediction_model, first_row)
print(predicted_price)
```

```
##          1
## 20169.36
```

Predict for all in cars93

```
predicted_prices <- predict(price_prediction_model, car_variables)
print(predicted_prices)
```

```
##          1          2          3          4          5          6          7          8
## 20169.363 38621.828 33073.148 33201.024 41899.199 22848.077 41868.559 61189.881
##          9         10         11         12         13         14         15         16
## 41975.122 53307.119 56734.423 21185.682 22443.134 37005.514 24190.781 42912.884
##         17         18         19         20         21         22         23         24
## 48234.520 53929.637 63949.203 36987.689 32023.435 36929.645 13428.896 21124.238
##         25         26         27         28         29         30         31         32
## 25290.623 34714.965 25759.503 42843.287 13535.459 41552.300 8506.048 18789.702
##         33         34         35         36         37         38         39         40
## 22204.527 23325.260 20669.587 34989.083 32997.701 51679.899 4930.298 15054.021
##         41         42         43         44         45         46         47         48
## 26070.298 14257.372 24992.503 13212.367 19027.090 13492.834 22049.248 55333.989
##         49         50         51         52         53         54         55         56
## 35979.899 37951.102 42340.160 53102.412 14024.660 17236.138 28410.447 35476.556
##         57         58         59         60         61         62         63         64
## 22341.201 24842.320 39301.342 15434.929 41002.898 13535.459 37746.364 16327.344
##         65         66         67         68         69         70         71         72
## 27220.194 36837.398 33439.827 26018.376 22890.702 42912.884 41868.559 17552.429
##         73         74         75         76         77         78         79         80
## 13741.245 19850.406 37005.514 39850.540 41975.122 23014.130 17441.924 8997.235
##         81         82         83         84         85         86         87         88
## 16815.551 24696.845 9358.785 12024.975 24365.137 24462.405 29768.202 15311.188
##         89         90         91         92         93
## 29949.274 22767.986 30957.290 24339.430 28928.842
```

Conclusion and Insights: The project successfully demonstrated the process of predicting car prices using linear regression model on two distinct datasets. By exploring various predictors, such as engine size, horsepower, and weight, we identified key factors influencing car prices. The analysis revealed: **Strong predictors:** Variables like horsepower and curb weight consistently showed a significant impact on car prices across both datasets. **Model performance:** Both models achieved reasonable accuracy, with R-squared values indicating a good fit. However, there is room for improvement in capturing price variability. **Dataset differences:** The comparison between the Cars93 and CarPrice_Assignment datasets highlighted differences in market segments and vehicle specifications. These findings provide valuable insights into how car features affect pricing, which could benefit both consumers and manufacturers. Future work could explore more advanced models or incorporate additional data to improve predictive power and better account for market dynamics.