

▼ EDA Using R Programming

Name: PALAK NATH

PRN: 17070124048



▼ DATASET : USED CAR ADVERTISMENT

Dataset contains 9276 rows and 10 variables with essential meanings:

- car: manufacturer brand
- price: seller's price in advertisement (in USD)
- body: car body type
- mileage: as mentioned in advertisement ('000 Km)
- engV: rounded engine volume ('000 cubic cm)
- engType: type of fuel ("Other" in this case should be treated as NA)
- registration: whether car registered in Ukraine or not
- year: year of production

- model: specific model name
- drive: drive type

```
sessionInfo()
```

```
R version 4.0.5 (2021-03-31)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Ubuntu 18.04.5 LTS

Matrix products: default
BLAS: /usr/lib/x86_64-linux-gnu/openblas/libblas.so.3
LAPACK: /usr/lib/x86_64-linux-gnu/libopenblas-p0.2.20.so
```

```
locale:
```

```
[1] LC_CTYPE=en_US.UTF-8      LC_NUMERIC=C
[3] LC_TIME=en_US.UTF-8       LC_COLLATE=en_US.UTF-8
[5] LC_MONETARY=en_US.UTF-8   LC_MESSAGES=en_US.UTF-8
[7] LC_PAPER=en_US.UTF-8     LC_NAME=C
[9] LC_ADDRESS=C              LC_TELEPHONE=C
[11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
```

```
attached base packages:
```

```
[1] stats      graphics    grDevices  utils      datasets  methods   base
```

```
loaded via a namespace (and not attached):
```

```
[1] fansi_0.4.2      digest_0.6.27    utf8_1.2.1      crayon_1.4.1
[5] IRdisplay_1.0    repr_1.1.3      lifecycle_1.0.0 jsonlite_1.7.2
[9] evaluate_0.14    pillar_1.5.1     rlang_0.4.10    uuid_0.1-4
[13] ellipsis_0.3.1  IRkernel_1.1.1   tools_4.0.5     compiler_4.0.5
[17] base64enc_0.1-3 pbdZMQ_0.3-5   htmltools_0.5.1.1
```

▼ INSTALLING LIBRARIES

```
install.packages("gplots")
install.packages("scales")
install.packages("ggplot2")
install.packages("hrbrthemes")
library(ggplot2) # Data visualization
https://colab.research.google.com/drive/1WqevASvzhap1HPRC_OTr2-ueny4n7L2#scrollTo=uChSLobjcf4q&printMode=true
```

```
library(readr) # CSV file I/O, e.g. the read_csv function  
library(gplots)  
library(ggplot2)  
library(repr)  
  
Installing package into ‘/usr/local/lib/R/site-library’  
(as ‘lib’ is unspecified)  
  
also installing the dependency ‘gdtools’  
  
Warning message in install.packages("hrbrthemes"):  
“installation of package ‘gdtools’ had non-zero exit status”  
Warning message in install.packages("hrbrthemes"):  
“installation of package ‘hrbrthemes’ had non-zero exit status”
```

▼ IMPORTING DATASETS

```
train <- read.csv("car_ad.csv")  
head(train)
```

A data.frame: 6 × 10

	car	price	body	mileage	engV	engType	registration	year	model	drive
	<chr>	<dbl>	<chr>	<int>	<dbl>	<chr>	<chr>	<int>	<chr>	<chr>
1	GAZ	259.35	sedan	1	2.4	Other	yes	1959	21	rear

▼ VARIABLE EXPLORATION

```
3           ZAZ 370.50 sedan      3    1.1 Petrol      yes 1989 968 rear
```

```
## show all variable names
```

```
var_name <- names(train)
```

```
var_name
```

```
'car' · 'price' · 'body' · 'mileage' · 'engV' · 'engType' · 'registration' · 'year' · 'model' · 'drive'
```

```
sapply(train, class)
```

```
car:      'character' price:      'numeric' body:      'character' mileage:      'integer' engV:      'numeric' engType:  

  'character' registration:      'character' year:      'integer' model:      'character' drive:      'character'
```

▼ REMOVING NOT REQUIRED COLUMNS

```
train = subset(train, select = -c(model) )  
train
```

A data.frame: 9309 × 9

	car	price	body	mileage	engV	engType	registration	year	drive
	<chr>	<dbl>	<chr>	<int>	<dbl>	<chr>	<chr>	<int>	<chr>
1	GAZ	259.350	sedan	1	2.40	Other	yes	1959	rear
2	Moskvich-AZLK	280.000	sedan	99	1.50	Petrol	yes	1976	rear
3	ZAZ	370.500	sedan	3	1.10	Petrol	yes	1989	rear
4	Moskvich-AZLK	400.000	sedan	1	10.00	Petrol	yes	1985	rear
5	ZAZ	400.000	sedan	20	NA	Petrol	yes	1990	rear
6	VAZ	400.000	sedan	10	1.10	Petrol	yes	1982	rear
7	VAZ	460.000	sedan	1	NA	Petrol	yes	1981	rear
8	VAZ	463.125	sedan	10	1.30	Petrol	yes	1979	rear
9	VAZ	463.125	sedan	68	1.20	Petrol	yes	1984	rear
10	GAZ	500.000	sedan	239	2.00	Other	yes	1961	rear
11	Opel	500.000	sedan	350	2.00	Petrol	yes	1990	rear
12	VAZ	537.225	sedan	87	1.50	Petrol	yes	1994	rear
13	Moskvich-AZLK	550.000	sedan	30	99.99	Petrol	yes	1971	rear
14	Toyota	600.000	sedan	10	1.50	Petrol	yes	1979	rear
15	VAZ	600.000	sedan	60	1.20	Petrol	yes	1976	rear
16	VAZ	649.000	sedan	9	1.20	Petrol	yes	1977	rear
17	VAZ	650.000	sedan	17	1.50	Petrol	yes	1988	rear
18	VAZ	650.000	sedan	10	1.30	Petrol	yes	1982	rear
19	VAZ	666.900	sedan	70	1.30	Petrol	yes	1983	rear
20	VAZ	670.000	sedan	5	1.50	Petrol	yes	1987	rear
21	Ford	680.000	sedan	100	2.30	Gas	yes	1980	rear

22	VAZ	700.000	vagon	1	NA	Petrol	yes	1981	rear
23	VAZ	703.950	sedan	187	1.60	Petrol	yes	1985	rear
24	VAZ	703.950	sedan	20	1.30	Petrol	yes	1985	rear
25	VAZ	703.950	sedan	30	1.30	Petrol	yes	1986	rear
26	VAZ	722.475	sedan	10	1.20	Petrol	yes	1984	rear
27	VAZ	741.000	sedan	167	1.30	Gas	yes	1986	rear
28	VAZ	750.000	sedan	264	1.50	Other	yes	1981	rear
29	VAZ	750.000	sedan	999	1.30	Petrol	yes	1984	rear
30	Moskvich-Izh	750.000	van	100	1.50	Petrol	yes	1988	rear
:	:	:	:	:	:	:	:	:	:
9280	Lexus	38700	crossover	32	NA	Petrol	yes	2013	
9281	BMW	39333	sedan	6	2.0	Petrol	yes	2016	
9282	Volkswagen	40000	sedan	1	1.4	Petrol	yes	2015	
9283	Volvo	40300	crossover	2	2.0	Diesel	yes	2016	
9284	Mercedes-Benz	41900	sedan	95	6.2	Petrol	yes	2011	
9285	Mercedes-Benz	41900	sedan	95	6.3	Other	yes	2011	
9286	Mercedes-Benz	41900	sedan	95	6.3	Petrol	yes	2011	
9287	Mercedes-Benz	42000	sedan	38	2.2	Diesel	yes	2014	
9288	Audi	44000	hatch	1	1.4	Petrol	yes	2016	
9289	Mercedes-Benz	45000	sedan	1	1.6	Petrol	yes	2015	
9290	Dodge	47000	other	14	6.4	Petrol	yes	2009	
9291	BMW	47900	crossover	178	4.0	Diesel	yes	2012	
9292	Audi	49500	crossover	44	3.0	Diesel	yes	2013	
9293	Porsche	51900	sedan	78	4.8	Petrol	yes	2011	

9294	Mercedes-Benz	55000	crossover	128	3.0	Diesel	yes	2013
9295	BMW	56500	crossover	1	2.0	Diesel	yes	2016
9296	BMW	57000	other	20	3.5	Petrol	yes	2013
9297	Mercedes-Benz	57000	sedan	43	4.7	Petrol	yes	2012
9298	BMW	57777	other	40	4.0	Diesel	yes	2012
9299	Ford	58915	crossover	0	NA	Other	yes	2016
9300	Audi	64400	other	41	NA	Other	ves	2014

▼ STATISTICAL SUMMARY

9302	BMW	77000	other	64	4.4	Petrol	ves	2012
------	-----	-------	-------	----	-----	--------	-----	------

summary(train)

car	price	body	mileage
Length:9309	Min. : 259.4	Length:9309	Min. : 0.0
Class :character	1st Qu.: 5400.0	Class :character	1st Qu.: 71.0
Mode :character	Median : 9500.0	Mode :character	Median :129.0
	Mean : 16081.7		Mean :140.3
	3rd Qu.: 17000.0		3rd Qu.:195.0
	Max. :547800.0		Max. :999.0

engV	engType	registration	year
Min. : 0.100	Length:9309	Length:9309	Min. :1953
1st Qu.: 1.600	Class :character	Class :character	1st Qu.:2004
Median : 2.000	Mode :character	Mode :character	Median :2008
Mean : 2.629			Mean :2007
3rd Qu.: 2.500			3rd Qu.:2012
Max. :99.990			Max. :2016

NA's :425

drive

Length:9309

Class :character

Mode :character

▼ DATA CLEANING : FINDING MISSING DATA

```
#Filling empty values with NA
train[train==""]<-NA

# As in Example one, you can create a data frame with logical TRUE and FALSE values;
# Indicating observed and missing values
is.na(train)
apply(is.na(train), 2, which)
```

A matrix: 9309 × 9 of type lgl

	car	price	body	mileage	engV	engType	registration	year	drive
1	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
2	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
3	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
4	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
5	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE
6	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
7	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE
8	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
9	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
10	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
11	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
12	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
13	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
14	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
15	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
16	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
17	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
18	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
19	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
20	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
21	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
22	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE

23	FALSE									
24	FALSE									
25	FALSE									
26	FALSE									
27	FALSE									
28	FALSE									
29	FALSE									
30	FALSE									
:	:	:	:	:	:	:	:	:	:	:
9280	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	TRUE	
9281	FALSE	TRUE								
9282	FALSE	TRUE								
9283	FALSE	TRUE								
9284	FALSE	TRUE								
9285	FALSE	TRUE								
9286	FALSE	TRUE								
9287	FALSE	TRUE								
9288	FALSE	TRUE								
9289	FALSE	TRUE								
9290	FALSE	TRUE								
9291	FALSE	TRUE								
9292	FALSE	TRUE								
9293	FALSE	TRUE								
9294	FALSE	TRUE								

```

9295 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE
9296 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE
9297 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE
9298 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE
9299 FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE TRUE
9300 FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE TRUE
9301 FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE TRUE
9302 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE
9303 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE
9304 FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE TRUE
9305 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE
9306 FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE TRUE
9307 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE
9308 FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE TRUE
9309 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE

```

\$car
\$price
\$body
\$mileage
\$engV

```

5:      5 7:      7 22:      22 39:      39 99:      99 113:      113 122:      122 269:      269 277:
      277 304:      304 327:      327 329:      329 397:      397 443:      443 506:      506 523:      523
      531:      531 545:      545 577:      577 596:      596 613:      613 627:      627 654:      654 656:
      656 778:      778 780:      780 807:      807 815:      815 842:      842 859:      859 893:      893
      930:      930 943:      943 955:      955 976:      976 990:      990 1026:      1026 1079:      1079 1144:
      1144 1151:      1151 1185:      1185 1196:      1196 1215:      1215 1216:      1216 1242:      1242
      1259:      1259 1302:      1302 1310:      1310 1314:      1314 1316:      1316 1317:      1317 1318:
      1318 1336:      1336 1338:      1338 1346:      1346 1352:      1352 1356:      1356 1358:      1358
      1361:      1361 1370:      1370 1376:      1376 1377:      1377 1380:      1380 1382:      1382 1389:
      1389 1400:      1400 1401:      1401 1413:      1413 1440:      1440 1446:      1446 1463:      1463
      1464:      1464 1465:      1465 1466:      1466 1467:      1467 1468:      1468 1469:      1469 1470:      1470 1471:
      1471 1472:      1472 1473:      1473 1474:      1474 1475:      1475 1476:      1476 1477:      1477 1478:      1478 1479:
      1479 1480:      1480 1481:      1481 1482:      1482 1483:      1483 1484:      1484 1485:      1485 1486:      1486 1487:
      1487 1488:      1488 1489:      1489 1490:      1490 1491:      1491 1492:      1492 1493:      1493 1494:      1494 1495:
      1495 1496:      1496 1497:      1497 1498:      1498 1499:      1499 1500:      1500 1501:      1501 1502:      1502 1503:
      1503 1504:      1504 1505:      1505 1506:      1506 1507:      1507 1508:      1508 1509:      1509 1510:      1510 1511:
      1511 1512:      1512 1513:      1513 1514:      1514 1515:      1515 1516:      1516 1517:      1517 1518:      1518 1519:
      1519 1520:      1520 1521:      1521 1522:      1522 1523:      1523 1524:      1524 1525:      1525 1526:      1526 1527:
      1527 1528:      1528 1529:      1529 1530:      1530 1531:      1531 1532:      1532 1533:      1533 1534:      1534 1535:
      1535 1536:      1536 1537:      1537 1538:      1538 1539:      1539 1540:      1540 1541:      1541 1542:      1542 1543:
      1543 1544:      1544 1545:      1545 1546:      1546 1547:      1547 1548:      1548 1549:      1549 1550:      1550 1551:
      1551 1552:      1552 1553:      1553 1554:      1554 1555:      1555 1556:      1556 1557:      1557 1558:      1558 1559:
      1559 1560:      1560 1561:      1561 1562:      1562 1563:      1563 1564:      1564 1565:      1565 1566:      1566 1567:
      1567 1568:      1568 1569:      1569 1570:      1570 1571:      1571 1572:      1572 1573:      1573 1574:      1574 1575:
      1575 1576:      1576 1577:      1577 1578:      1578 1579:      1579 1580:      1580 1581:      1581 1582:      1582 1583:
      1583 1584:      1584 1585:      1585 1586:      1586 1587:      1587 1588:      1588 1589:      1589 1590:      1590 1591:
      1591 1592:      1592 1593:      1593 1594:      1594 1595:      1595 1596:      1596 1597:      1597 1598:      1598 1599:
      1599 1600:      1600 1601:      1601 1602:      1602 1603:      1603 1604:      1604 1605:      1605 1606:      1606 1607:
      1607 1608:      1608 1609:      1609 1610:      1610 1611:      1611 1612:      1612 1613:      1613 1614:      1614 1615:
      1615 1616:      1616 1617:      1617 1618:      1618 1619:      1619 1620:      1620 1621:      1621 1622:      1622 1623:
      1623 1624:      1624 1625:      1625 1626:      1626 1627:      1627 1628:      1628 1629:      1629 1630:      1630 1631:
      1631 1632:      1632 1633:      1633 1634:      1634 1635:      1635 1636:      1636 1637:      1637 1638:      1638 1639:
      1639 1640:      1640 1641:      1641 1642:      1642 1643:      1643 1644:      1644 1645:      1645 1646:      1646 1647:
      1647 1648:      1648 1649:      1649 1650:      1650 1651:      1651 1652:      1652 1653:      1653 1654:      1654 1655:
      1655 1656:      1656 1657:      1657 1658:      1658 1659:      1659 1660:      1660 1661:      1661 1662:      1662 1663:
      1663 1664:      1664 1665:      1665 1666:      1666 1667:      1667 1668:      1668 1669:      1669 1670:      1670 1671:
      1671 1672:      1672 1673:      1673 1674:      1674 1675:      1675 1676:      1676 1677:      1677 1678:      1678 1679:
      1679 1680:      1680 1681:      1681 1682:      1682 1683:      1683 1684:      1684 1685:      1685 1686:      1686 1687:
      1687 1688:      1688 1689:      1689 1690:      1690 1691:      1691 1692:      1692 1693:      1693 1694:      1694 1695:
      1695 1696:      1696 1697:      1697 1698:      1698 1699:      1699 1700:      1700 1701:      1701 1702:      1702 1703:
      1703 1704:      1704 1705:      1705 1706:      1706 1707:      1707 1708:      1708 1709:      1709 1710:      1710 1711:
      1711 1712:      1712 1713:      1713 1714:      1714 1715:      1715 1716:      1716 1717:      1717 1718:      1718 1719:
      1719 1720:      1720 1721:      1721 1722:      1722 1723:      1723 1724:      1724 1725:      1725 1726:      1726 1727:
      1727 1728:      1728 1729:      1729 1730:      1730 1731:      1731 1732:      1732 1733:      1733 1734:      1734 1735:
      1735 1736:      1736 1737:      1737 1738:      1738 1739:      1739 1740:      1740 1741:      1741 1742:      1742 1743:
      1743 1744:      1744 1745:      1745 1746:      1746 1747:      1747 1748:      1748 1749:      1749 1750:      1750 1751:
      1751 1752:      1752 1753:      1753 1754:      1754 1755:      1755 1756:      1756 1757:      1757 1758:      1758 1759:
      1759 1760:      1760 1761:      1761 1762:      1762 1763:      1763 1764:      1764 1765:      1765 1766:      1766 1767:
      1767 1768:      1768 1769:      1769 1770:      1770 1771:      1771 1772:      1772 1773:      1773 1774:      1774 1775:
      1775 1776:      1776 1777:      1777 1778:      1778 1779:      1779 1780:      1780 1781:      1781 1782:      1782 1783:
      1783 1784:      1784 1785:      1785 1786:      1786 1787:      1787 1788:      1788 1789:      1789 1790:      1790 1791:
      1791 1792:      1792 1793:      1793 1794:      1794 1795:      1795 1796:      1796 1797:      1797 1798:      1798 1799:
      1799 1800:      1800 1801:      1801 1802:      1802 1803:      1803 1804:      1804 1805:      1805 1806:      1806 1807:
      1807 1808:      1808 1809:      1809 1810:      1810 1811:      1811 1812:      1812 1813:      1813 1814:      1814 1815:
      1815 1816:      1816 1817:      1817 1818:      1818 1819:      1819 1820:      1820 1821:      1821 1822:      1822 1823:
      1823 1824:      1824 1825:      1825 1826:      1826 1827:      1827 1828:      1828 1829:      1829 1830:      1830 1831:
      1831 1832:      1832 1833:      1833 1834:      1834 1835:      1835 1836:      1836 1837:      1837 1838:      1838 1839:
      1839 1840:      1840 1841:      1841 1842:      1842 1843:      1843 1844:      1844 1845:      1845 1846:      1846 1847:
      1847 1848:      1848 1849:      1849 1850:      1850 1851:      1851 1852:      1852 1853:      1853 1854:      1854 1855:
      1855 1856:      1856 1857:      1857 1858:      1858 1859:      1859 1860:      1860 1861:      1861 1862:      1862 1863:
      1863 1864:      1864 1865:      1865 1866:      1866 1867:      1867 1868:      1868 1869:      1869 1870:      1870 1871:
      1871 1872:      1872 1873:      1873 1874:      1874 1875:      1875 1876:      1876 1877:      1877 1878:      1878 1879:
      1879 1880:      1880 1881:      1881 1882:      1882 1883:      1883 1884:      1884 1885:      1885 1886:      1886 1887:
      1887 1888:      1888 1889:      1889 1890:      1890 1891:      1891 1892:      1892 1893:      1893 1894:      1894 1895:
      1895 1896:      1896 1897:      1897 1898:      1898 1899:      1899 1900:      1900 1901:      1901 1902:      1902 1903:
      1903 1904:      1904 1905:      1905 1906:      1906 1907:      1907 1908:      1908 1909:      1909 1910:      1910 1911:
      1911 1912:      1912 1913:      1913 1914:      1914 1915:      1915 1916:      1916 1917:      1917 1918:      1918 1919:
      1919 1920:      1920 1921:      1921 1922:      1922 1923:      1923 1924:      1924 1925:      1925 1926:      1926 1927:
      1927 1928:      1928 1929:      1929 1930:      1930 1931:      1931 1932:      1932 1933:      1933 1934:      1934 1935:
      1935 1936:      1936 1937:      1937 1938:      1938 1939:      1939 1940:      1940 1941:      1941 1942:      1942 1943:
      1943 1944:      1944 1945:      1945 1946:      1946 1947:      1947 1948:      1948 1949:      1949 1950:      1950 1951:
      1951 1952:      1952 1953:      1953 1954:      1954 1955:      1955 1956:      1956 1957:      1957 1958:      1958 1959:
      1959 1960:      1960 1961:      1961 1962:      1962 1963:      1963 1964:      1964 1965:      1965 1966:      1966 1967:
      1967 1968:      1968 1969:      1969 1970:      1970 1971:      1971 1972:      1972 1973:      1973 1974:      1974 1975:
      1975 1976:      1976 1977:      1977 1978:      1978 1979:      1979 1980:      1980 1981:      1981 1982:      1982 1983:
      1983 1984:      1984 1985:      1985 1986:      1986 1987:      1987 1988:      1988 1989:      1989 1990:      1990 1991:
      1991 1992:      1992 1993:      1993 1994:      1994 1995:      1995 1996:      1996 1997:      1997 1998:      1998 1999:
      1999 2000:      2000 2001:      2001 2002:      2002 2003:      2003 2004:      2004 2005:      2005 2006:      2006 2007:
      2007 2008:      2008 2009:      2009 2010:      2010 2011:      2011 2012:      2012 2013:      2013 2014:      2014 2015:
      2015 2016:      2016 2017:      2017 2018:      2018 2019:      2019 2020:      2020 2021:      2021 2022:      2022 2023:
      2023 2024:      2024 2025:      2025 2026:      2026 2027:      2027 2028:      2028 2029:      2029 2030:      2030 2031:
      2031 2032:      2032 2033:      2033 2034:      2034 2035:      2035 2036:      2036 2037:      2037 2038:      2038 2039:
      2039 2040:      2040 2041:      2041 2042:      2042 2043:      2043 2044:      2044 2045:      2045 2046:      2046 2047:
      2047 2048:      2048 2049:      2049 2050:      2050 2051:      2051 2052:      2052 2053:      2053 2054:      2054 2055:
      2055 2056:      2056 2057:      2057 2058:      2058 2059:      2059 2060:      2060 2061:      2061 2062:      2062 2063:
      2063 2064:      2064 2065:      2065 2066:      2066 2067:      2067 2068:      2068 2069:      2069 2070:      2070 2071:
      2071 2072:      2072 2073:      2073 2074:      2074 2075:      2075 2076:      2076 2077:      2077 2078:      2078 2079:
      2079 2080:      2080 2081:      2081 2082:      2082 2083:      2083 2084:      2084 2085:      2085 2086:      2086 2087:
      2087 2088:      2088 2089:      2089 2090:      2090 2091:      2091 2092:      2092 2093:      2093 2094:      2094 2095:
      2095 2096:      2096 2097:      2097 2098:      2098 2099:      2099 2100:      2100 2101:      2101 2102:      2102 2103:
      2103 2104:      2104 2105:      2105 2106:      2106 2107:      2107 2108:      2108 2109:      2109 2110:      2110 2111:
      2111 2112:      2112 2113:      2113 2114:      2114 2115:      2115 2116:      2116 2117:      2117 2118:      2118 2119:
      2119 2120:      2120 2121:      2121 2122:      2122 2123:      2123 2124:      2124 2125:      2125 2126:      2126 2127:
      2127 2128:      2128 2129:      2129 2130:      2130 2131:      2131 2132:      2132 2133:      2133 2134:      2134 2135:
      2135 2136:      2136 2137:      2137 2138:      2138 2139:      2139 2140:      2140 2141:      2141 2142:      2142 2143:
      2143 2144:      2144 2145:      2145 2146:      2146 2147:      2147 2148:      2148 2149:      2149 2150:      2150 2151:
      2151 2152:      2152 2153:      2153 2154:      2154 2155:      2155 2156:      2156 2157:      2157 2158:      2158 2159:
      2159 2160:      2160 2161:      2161 2162:      2162 2163:      2163 2164:      2164 2165:      2165 2166:      2166 2167:
      2167 2168:      2168 2169:      2169 2170:      2170 2171:      2171 2172:      2172 2173:      2173 2174:      2174 2175:
      2175 2176:      2176 2177:      2177 2178:      2178 2179:      2179 2180:      2180 2181:      2181 2182:      2182 2183:
      2183 2184:      2184 2185:      2185 2186:      2186 2187:      2187 2188:      2188 2189:      2189 2190:      2190 2191:
      2191 2192:      2192 2193:      2193 2194:      2194 2195:      2195 2196:      2196 2197:      2197 2198:      2198 2199:
      2199 2200:      2200 2201:      2201 2202:      2202 2203:      2203 2204:      2204 2205:      2205 2206:      2206 2207:

```

```

1482: 1482 1499: 1499 1503: 1503 1567: 1567 1573: 1573 1576: 1576 1599:
      1599 1630: 1630 1671: 1671 1707: 1707 1771: 1771 1775: 1775 1800: 1800
1863: 1863 1911: 1911 1934: 1934 1979: 1979 2040: 2040 2073: 2073 2097:
      2097 2109: 2109 2123: 2123 2138: 2138 2167: 2167 2193: 2193 2345:
2349: 2349 2359: 2359 2376: 2376 2425: 2425 2488: 2488 2492: 2492 2545:
      2545 2560: 2560 2684: 2684 2857: 2857 2891: 2891 2923: 2923 3029: 3029
3158: 3158 3175: 3175 3278: 3278 3386: 3386 3583: 3583 3676: 3676 3712:
      3712 3720: 3720 3724: 3724 3733: 3733 3736: 3736 3737: 3737 3765:
3766: 3766 3771: 3771 3778: 3778 3782: 3782 3783: 3783 3786: 3786 3788:
      3788 3792: 3792 3806: 3806 3822: 3822 3851: 3851 3858: 3858 3870: 3870
3879: 3879 3893: 3893 3911: 3911 3941: 3941 3979: 3979 3983: 3983 3989:
      3989 4085: 4085 4133: 4133 4175: 4175 4221: 4221 4235: 4235 4302: 4302
4327: 4327 4334: 4334 4343: 4343 4352: 4352 4355: 4355 4357: 4357 4379:
      4379 4384: 4384 4400: 4400 4402: 4402 4492: 4492 4505: 4505 4517: 4517
4541: 4541 4547: 4547 4580: 4580 4596: 4596 4609: 4609 4630: 4630 4638:

```

▼ HANDLING MISSING DATA

Missing Data is only in the column, engV so replacing the NA values using mean

```
5575 5550. 5555 5549. 5549 5554. 5554 5557. 5557 5572. 5572 5585. ...
```

#Replacing Values with mean in engV

```
train$engV[is.na(train$engV)]<-mean(train$engV,na.rm=TRUE)
```

```
----- 1700 5720. 1720 5720. 1700 5730. 1742 5710. 1742 5714. 1701 5720. 1701
```

#Removing rows with NA in drive Column

```
train <- na.omit(train) # remove na in r - remove rows - na.omit function / option
```

```
----- 7707 2020 7075 2020 7001 2020 7010 2020 7000 2020 7000 2020 7074 2020
```

▼ CHECK FOR NA VALUES AGAIN

```
8381 8188: 8382 8193: 8399 8216: 8402 8217: 8403 8223: 8408 8228: 8411
```

```
apply(is.na(train), 2, which)
```

#will not show anything as no NA Value present now

```
8860 8470: 8860 8470: 8891 8470: 8893 8470: 8897 8497: 8898 8517: 8899
```

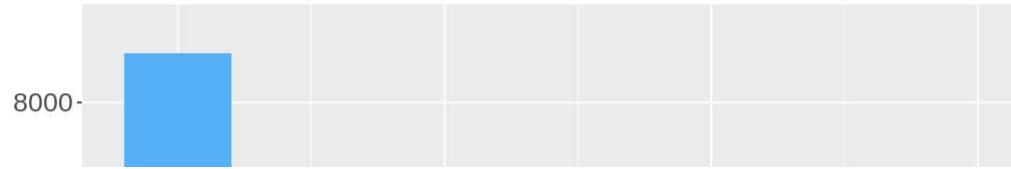
▼ VISUALISATION

```
8943 8893: 8957 8897: 8964 8898: 8968 8899: 8969 8901: 8974 8904: 8982
```

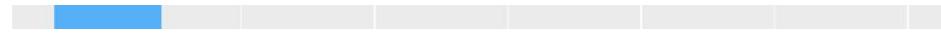
HISTOGRAM

```
# Draw a histogram to figure out the distribution of SalePrice
options(scipen=500)
options(repr.plot.width=10, repr.plot.height=10)
ggplot(train, aes(x = price, fill = ..count..)) +
  geom_histogram(binwidth = 80000) +
  ggtitle("Figure 1 Histogram of SalePrice") +
  ylab("Count of Cars") +
  xlab("Car Price") +
  theme(text = element_text(size = 20)) +
  theme(plot.title = element_text(hjust = 0.5, size = 30))
```

Figure 1 Histogram of SalePrice

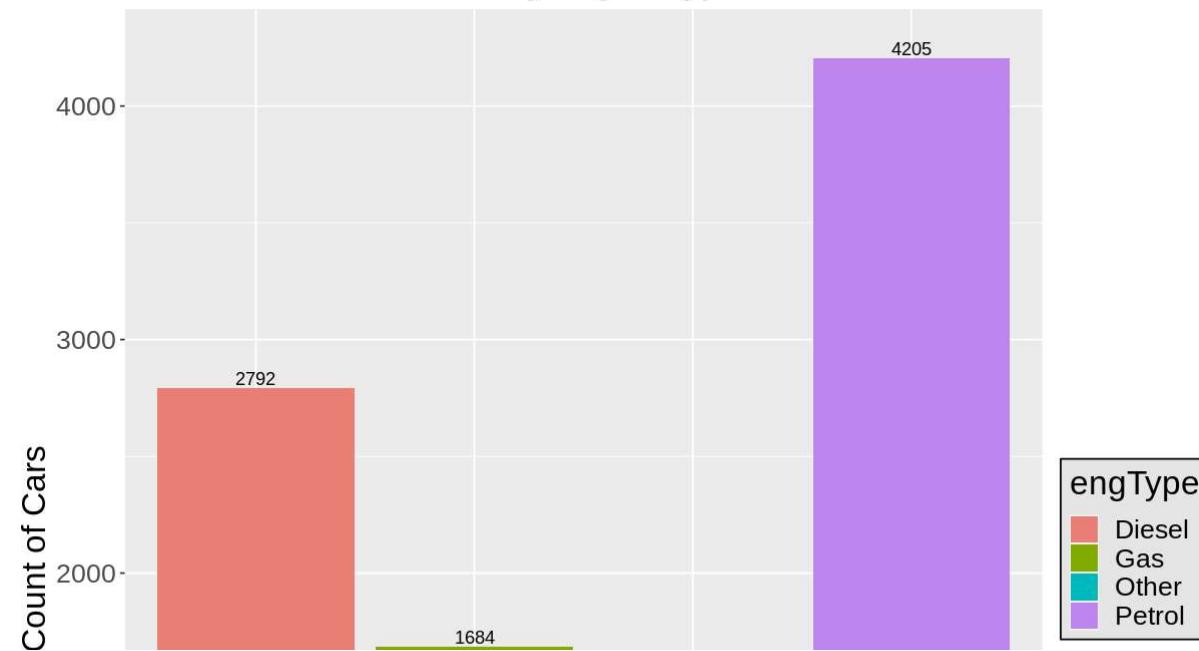


BAR CHART



```
#Cars by Engine Type
options(repr.plot.width=10, repr.plot.height=10)
bp1 <- ggplot(train, aes(x = engType, fill = engType )) +
geom_bar()+
scale_fill_hue(c = 80)+ 
ggtitle("Cars by Engine Type")+
ylab("Count of Cars") +
xlab("Engine Type") +
theme(plot.title = element_text(hjust = 0.5, size=20), legend.position="right", legend.background = element_rect(fill="grey90",
size=0.5, linetype="solid",
colour ="black"))+
geom_text(stat='count',aes(label=..count..),vjust=-0.25) +
theme(text = element_text(size = 20))
bp1
```

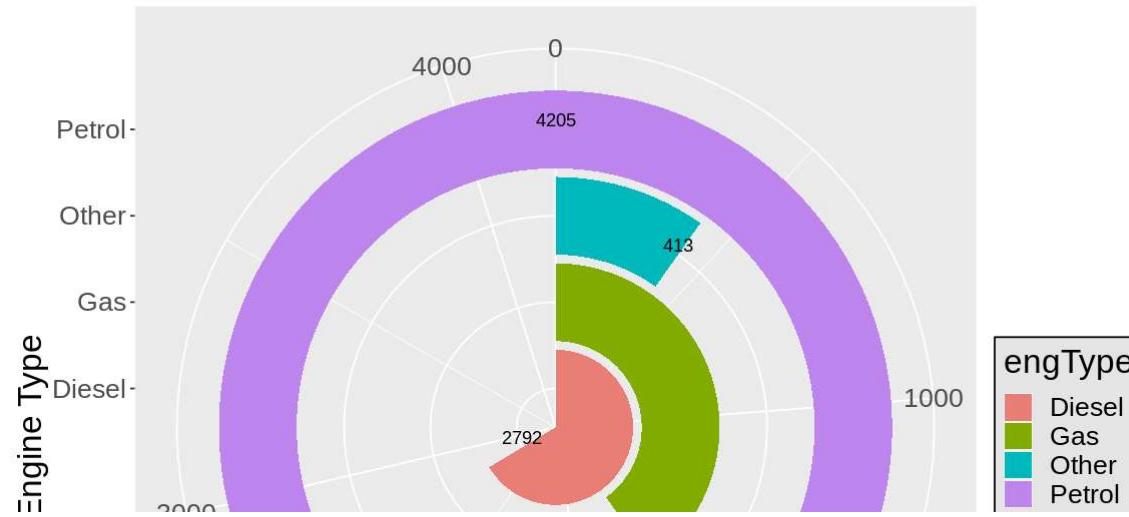
Cars by Engine Type



PIE CHART

```
options(repr.plot.width=10, repr.plot.height=8)
pie <- bp1 + coord_polar("y")+
theme(text = element_text(size = 20)) +
theme(plot.title = element_text(hjust = 0.5, size = 30))
pie
```

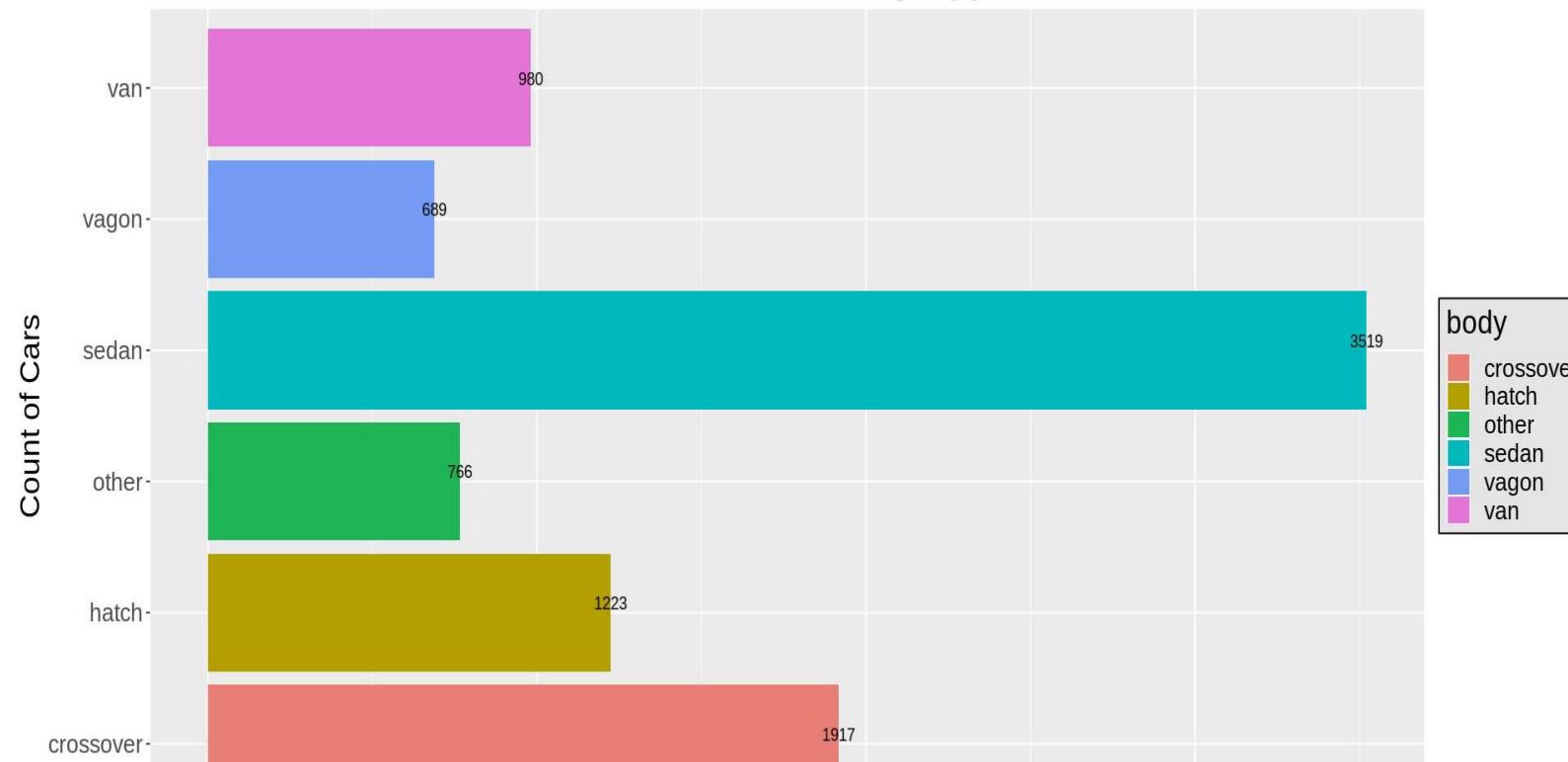
Cars by Engine Type



HORIZONTAL BAR CHART

```
# count car by body
options(repr.plot.width=16, repr.plot.height=8)
bp <- ggplot(train, aes(x = body, fill = body )) +
  geom_bar()+
  scale_fill_hue(c = 80)+ 
  ggttitle("Distribution of Body Type")+
  ylab("Body of Car") +
  xlab("Count of Cars") +
  theme(text = element_text(size = 20)) +
  theme(plot.title = element_text(hjust = 0.5,size = 30),legend.position="right", legend.background = element_rect(fill="grey90",
                                                                 size=0.5, linetype="solid",
                                                                 colour ="black"))+
  geom_text(stat='count',aes(label=..count..),vjust=-0.25) +
  coord_flip()
bp
```

Distribution of Body Type



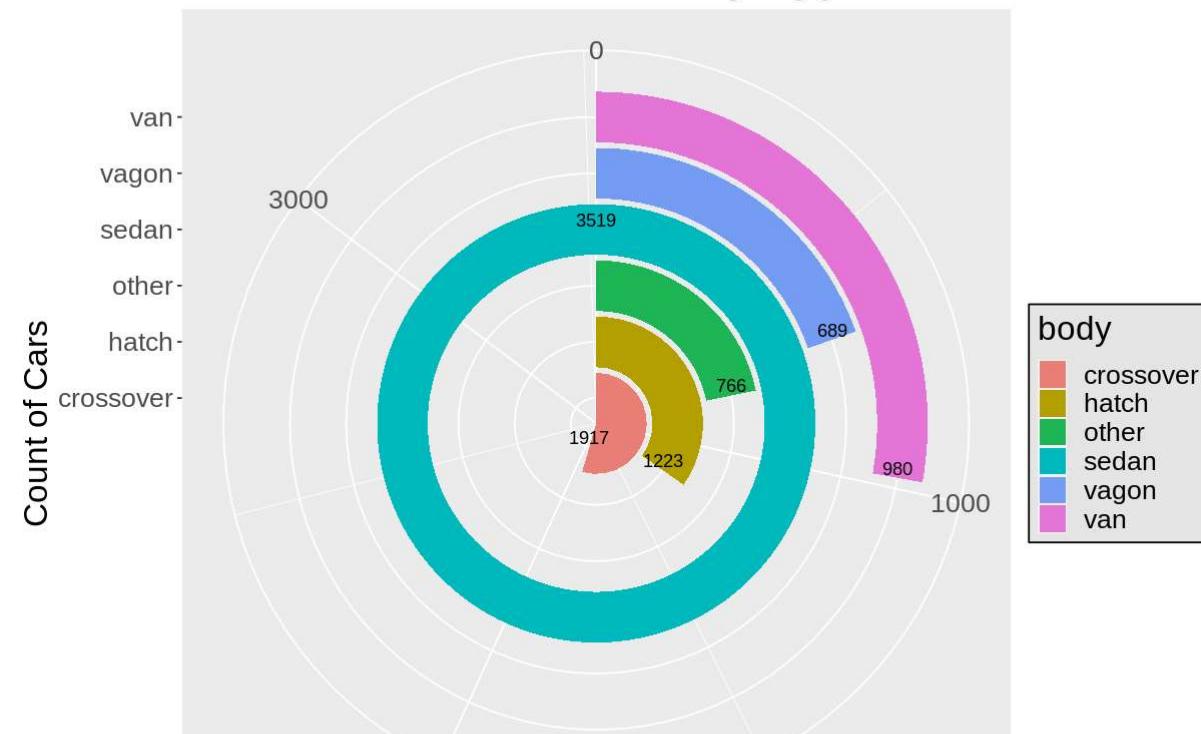
```
# Distribution of body type
table(train$body)
```

body	Count
crossover	1917
hatch	1223
other	766
sedan	3519
vagon	689
van	980

```
options(repr.plot.width=10, repr.plot.height=8)
pie <- bp + coord_polar("y", start=0)
pie
```

Coordinate system already present. Adding new coordinate system, which will replace the existing one.

Distribution of Body Type

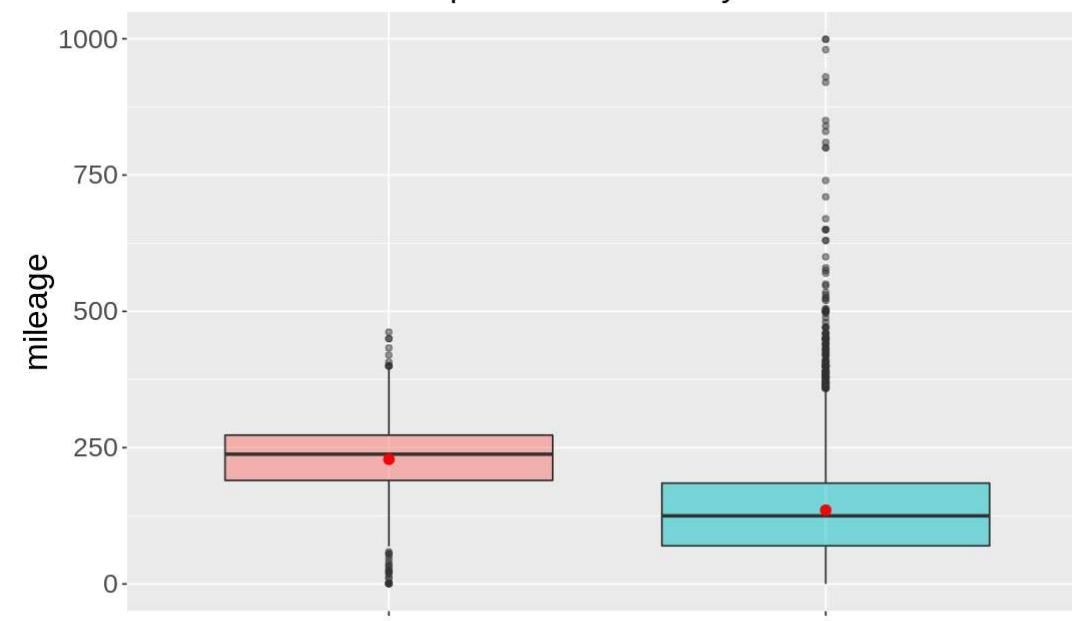


```
# Change plot size to 9 x 6
options(repr.plot.width=9, repr.plot.height=6)
#boxplot of SalePrice by MSZoning
#add average value of SalePrice as red point
ggplot(train, aes(x=registration, y=mileage, fill=registration)) +
  geom_boxplot(alpha=0.5) +
  theme(text = element_text(size = 20)) +
  theme(plot.title = element_text(hjust = 0.5, size = 30)) +
  stat_summary(fun.y=mean, geom="point", shape=20, size=4, color="red", fill="red")+
  theme(legend.position="none")+
  ggtitle("Boxplot of SalePrice by Price")+
  theme(plot.title = element_text(hjust = 0.5, size=20))
```

Warning message:

"`fun.y` is deprecated. Use `fun` instead."

Boxplot of SalePrice by Price



```
# Change plot size to 9 x 6
options(repr.plot.width=9, repr.plot.height=6)
#boxplot of Mileage by MSZoning
#add average value of Mileage as red point
ggplot(train, aes(x=drive, y=mileage, fill=drive)) +
  theme(text = element_text(size = 20)) +
  theme(plot.title = element_text(hjust = 0.5, size = 30)) +
  geom_boxplot(alpha=0.5) +
  stat_summary(fun.y=mean, geom="point", shape=20, size=4, color="red", fill="red")+
  theme(legend.position="none")+
  ggtitle("Boxplot of SalePrice by Price")+
  theme(plot.title = element_text(hjust = 0.5))
```

Warning message:

“`fun.y` is deprecated. Use `fun` instead.”

Boxplot of SalePrice by Price



```
library(plyr)
ddply(train, .(drive), summarize, Total = length(drive), Max_price=max(price), Min_price=min(price))
```

Error in library(plyr): there is no package called ‘plyr’
Traceback:

```
1. library(plyr)
```

SEARCH STACK OVERFLOW

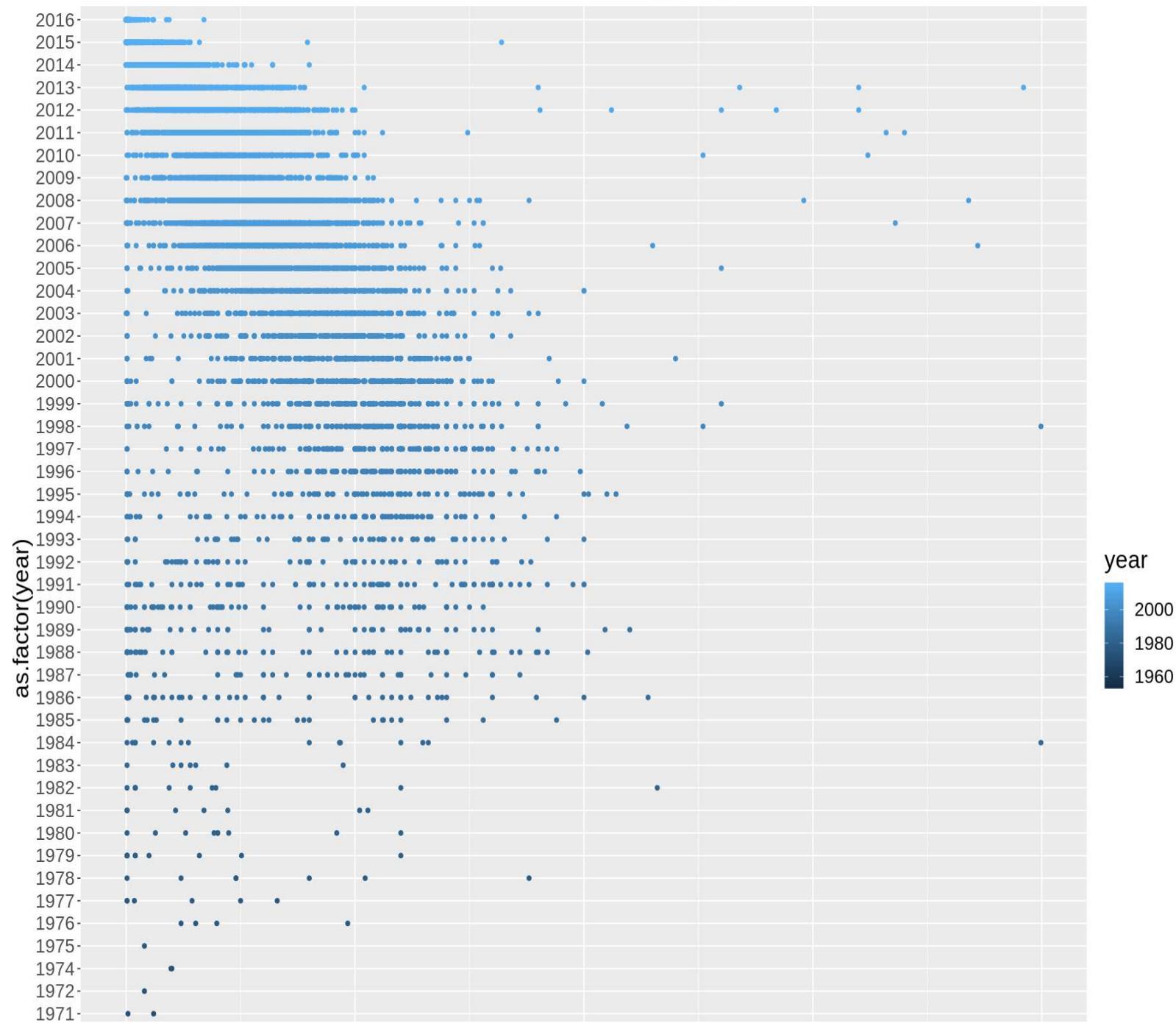
```
options(repr.plot.width=10, repr.plot.height=8)
ggplot(train, aes(price, as.factor(body), col= body)) +
theme(text = element_text(size = 20)) +
theme(plot.title = element_text(hjust = 0.5, size = 30)) +
ggttitle("Scatterplot of SalePrice as a factor of Body")+
geom_point()
```

Scatterplot of SalePrice as a factor of Body



```
options(repr.plot.width=15, repr.plot.height=15)
ggplot(train, aes(mileage, as.factor(year), col= year)) +
  ggtitle("Scatter Plot of Mileage by Year") +
  theme(text = element_text(size = 20)) +
  theme(plot.title = element_text(hjust = 0.5, size = 30)) +
  geom_point()
```

Scatter Plot of Mileage by Year



```
options(repr.plot.width=15, repr.plot.height=10)
plot(x=train$year, y=train$price, pch=18,
      cex=2,
      col="#69b3a2",
      cex.main=2,
      cex.lab = 1.5,
      cex.axis=1,
      xlab="Year", ylab="Price",
      main="Scatterplot showing Price of the used car vs the year")
```

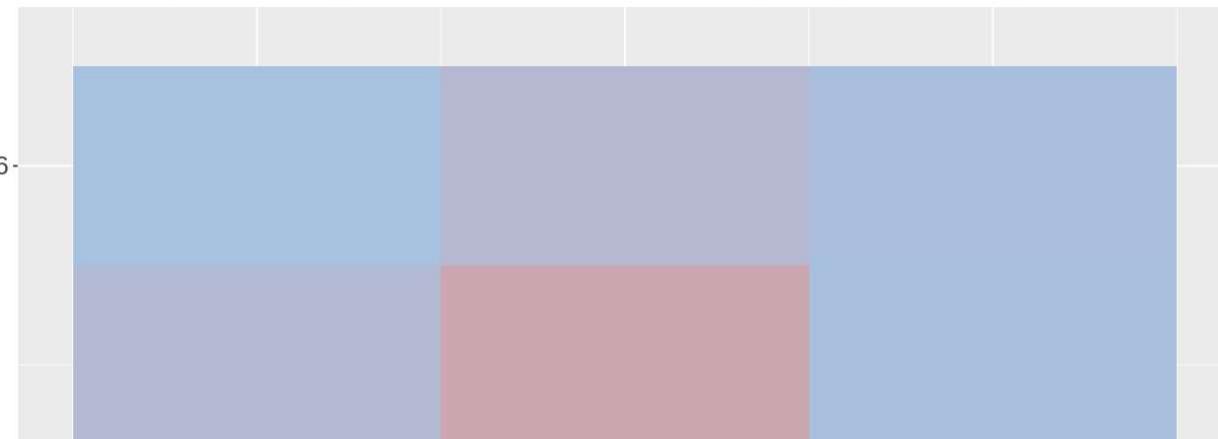
Scatterplot showing Price of the used car vs the year

```
options(repr.plot.width=12, repr.plot.height=12)
# Heatmap
ggplot(train, aes(drive, body, fill= mileage)) +
geom_tile()+
theme(text = element_text(size = 20)) +
ggtitle("Heat Map showing relation between Drive,Body & Mileage")+
scale_fill_gradient(low="lightskyblue", high="red")
```

Warning message:

“Removed 16 rows containing missing values (geom_tile).”

Heat Map showing relation between Drive,Body & Mileage



```
options(repr.plot.width=12, repr.plot.height=12)
# Heatmap
ggplot(train, aes(body, engType, fill= mileage)) +
  theme(text = element_text(size = 20)) +
  geom_tile()+
  ggtitle("Heat Map showing relation between engType,Body & Mileage")+
  scale_fill_gradient(low="olivedrab2", high="black")
```

Heat Map showing relation between engType,Body & Mileage



▼ CONVERTING CATEGORICAL TO NUMERICAL DATA

```
dataInfo <- function(train) {  
  NAvalues <- list(); NAclass <- list(); UniqueVals <- list(); ColName <- names(train); Index <- c(1:dim(train)[2])  
  for(i in 1:dim(train)[2]) {  
    NAvalues[i] <- sum(is.na(train[,i]))  
    NAclass[i] <- class(train[,i])  
    if(class(train[,i])=="numeric" | class(train[,i])=="integer") {  
      UniqueVals[i] <- 'Not Applicable'  
    }  
    else{  
      UniqueVals[i] <- length(unique(train[,i]))  
    }  
  }  
}
```

```

        NAlist <- cbind(Index,ColName,NAvalues,NAclass,UniqueVals)
    }

Info <- dataInfo(train);
Info

```

A matrix: 9 × 5

Index	ColName	NAvalues	NAclass	UniqueVals
<int>	<int>	<int>	<int>	<int> <int> <int> <int> <int> <chr> ... <chr> <int> <chr> <int> <chr> <int> <chr> <int> <int> <chr> <int>
1	car	0	character	85
2	price	0	numeric	Not Applicable
3	body	0	character	6
4	mileage	0	integer	Not Applicable
5	engV	0	numeric	Not Applicable
6	engType	0	character	4
7	registration	0	character	2

```

# convert factor to numeric
train$drive <- as.numeric(factor(train$drive,
                                levels = c("front", "full", "rear", " "),
                                labels = c(1,2,3,4) ,ordered = TRUE))
train$engType <- as.numeric(factor(train$engType,
                                levels = c("Petrol", "Diesel","Gas", "Other"),
                                labels = c(1,2,3,4) ,ordered = TRUE))
train$registration <- as.numeric(factor(train$registration,
                                levels = c("N", "Y"),
                                labels = c(0,1) ,ordered = TRUE))
train$body <- as.numeric(factor(train$body,
                                levels = c("crossover","hatch", "sedan", "wagon", "van","other"),
                                labels = c(1,2,3,4,5,6) ,ordered = TRUE))

```

train

A data.frame: 9094 × 9

	car	price	body	mileage	engV	engType	registration	year	drive
	<chr>	<dbl>	<dbl>	<int>	<dbl>	<dbl>	<dbl>	<int>	<dbl>
1	GAZ	259.350	3	1	2.400000	4	NA	1959	3
2	Moskvich-AZLK	280.000	3	99	1.500000	1	NA	1976	3
3	ZAZ	370.500	3	3	1.100000	1	NA	1989	3
4	Moskvich-AZLK	400.000	3	1	10.000000	1	NA	1985	3
5	ZAZ	400.000	3	20	2.629093	1	NA	1990	3
6	VAZ	400.000	3	10	1.100000	1	NA	1982	3
7	VAZ	460.000	3	1	2.629093	1	NA	1981	3
8	VAZ	463.125	3	10	1.300000	1	NA	1979	3
9	VAZ	463.125	3	68	1.200000	1	NA	1984	3
10	GAZ	500.000	3	239	2.000000	4	NA	1961	3
11	Opel	500.000	3	350	2.000000	1	NA	1990	3
12	VAZ	537.225	3	87	1.500000	1	NA	1994	3
13	Moskvich-AZLK	550.000	3	30	99.990000	1	NA	1971	3
14	Toyota	600.000	3	10	1.500000	1	NA	1979	3
15	VAZ	600.000	3	60	1.200000	1	NA	1976	3
16	VAZ	649.000	3	9	1.200000	1	NA	1977	3
17	VAZ	650.000	3	17	1.500000	1	NA	1988	3
18	VAZ	650.000	3	10	1.300000	1	NA	1982	3
19	VAZ	666.900	3	70	1.300000	1	NA	1983	3
20	VAZ	670.000	3	5	1.500000	1	NA	1987	3
21	Ford	680.000	3	100	2.300000	3	NA	1980	3

22	VAZ	700.000	4	1	2.629093	1	NA	1981	3
23	VAZ	703.950	3	187	1.600000	1	NA	1985	3
24	VAZ	703.950	3	20	1.300000	1	NA	1985	3
25	VAZ	703.950	3	30	1.300000	1	NA	1986	3
26	VAZ	722.475	3	10	1.200000	1	NA	1984	3
27	VAZ	741.000	3	167	1.300000	3	NA	1986	3
28	VAZ	750.000	3	264	1.500000	4	NA	1981	3
29	VAZ	750.000	3	999	1.300000	1	NA	1984	3
30	Moskvich-lzh	750.000	5	100	1.500000	1	NA	1988	3
:	:	:	:	:	:	:	:	:	:
9065	Renault	7250	5	83	2.629093	4	NA	2013	1
9066	Kia	7250	3	170	1.600000	2	NA	2008	1
9067	Skoda	7300	6	0	2.629093	4	NA	2012	1
9068	Mitsubishi	7300	2	126	2.629093	4	NA	2004	1
9069	Mazda	7500	3	170	2.000000	1	NA	2005	1
9070	Smart	7500	6	150	0 700000	1	NA	2006	1

END OF ASSIGNMENT

9072	Skoda	8200	6	147	1.600000	3	NA	2007	1
9073	Hyundai	8200	2	27	1.400000	1	NA	2011	1
9074	Skoda	8200	3	1	18.000000	3	NA	2005	1
9075	Renault	8250	3	148	2.629093	4	NA	2011	1
9076	Toyota	8300	3	140	3.000000	1	NA	2002	1
9077	Opel	8500	5	230	1.900000	2	NA	2004	1
9078	Peugeot	8500	5	180	1.600000	2	NA	2007	1

9079	Ford	8800	2	16	1.250000	1	NA	2014	1
9080	Hyundai	8800	2	122	1.600000	2	NA	2008	1
9081	BMW	8900	3	256	2.200000	1	NA	2002	1
9082	Geely	8900	2	35	1.800000	3	NA	2014	1
9083	Chrysler	8900	3	236	2.400000	3	NA	2008	1
9084	Hyundai	8999	2	74	1.400000	1	NA	2011	1

✓ 0s completed at 3:06 PM

