

# Final\_project EES6690 Columbia University – Haoran Guo (hg2461@columbia.edu), Hao Fang (hg2345@columbia.edu)

Notice: This work is the final project of course EECS E6690 TPC: Statistical Learning in Bio & Info System. Prof. Predrag R. Jelenkovi ??c. Instructors: Ludwig Zhao, Tingkai Liu.

This work aims to reproduce the result in article: Beata Strack,1 Jonathan P. DeShazo Impact of HbA1c Measurement on Hospital Readmission Rates: Analysis of 70,000 Clinical Database Patient Records, 2014. Print. Major part includes training, evaluating and improving a logistic model. Some statistical graphs are then generated for analysis purpose.

Data set: Diabetes 130-US hospitals for years 1999-2008 Data Set

Loading libraries

```
library(GGally)
library(ggplot2)
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
library(psych)
```

```
##
```

```
## Attaching package: 'psych'
```

```
## The following objects are masked from 'package:ggplot2':
```

```
##
```

```
##      %+%, alpha
```

```
library(lasso2)
```

```
## R Package to solve regression problems while imposing
```

```
##   an L1 constraint on the parameters. Based on S-plus Release 2.1
```

```
## Copyright (C) 1998, 1999
```

```
## Justin Lokhorst <jlokhors@stats.adelaide.edu.au>
```

```
## Berwin A. Turlach <bturlach@stats.adelaide.edu.au>
```

```
## Bill Venables <wvenable@stats.adelaide.edu.au>
```

```
##
```

```
## Copyright (C) 2002
```

```
## Martin Maechler <maechler@stat.math.ethz.ch>
```

```
##
```

```
## Attaching package: 'lasso2'
```

```
## The following object is masked from 'package:psych':
```

```
##
```

```
##      tr
```

```
library(dplyr)
```

```
##
```

```
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:GGally':
```

```

##
##      nasa
## The following objects are masked from 'package:stats':
##
##      filter, lag
## The following objects are masked from 'package:base':
##
##      intersect, setdiff, setequal, union
library(proto)
library(RSQLite)
library(gsubfn)
library(sqldf)
library(lattice)
library(survival)
library(grid)
library(Matrix)
library(survey)

##
## Attaching package: 'survey'
## The following object is masked from 'package:graphics':
##
##      dotchart
library(caret)

##
## Attaching package: 'caret'
## The following object is masked from 'package:survival':
##
##      cluster
library(rpart)
library(randomForest)

## randomForest 4.6-14
## Type rfNews() to see new features/changes/bug fixes.
##
## Attaching package: 'randomForest'
## The following object is masked from 'package:dplyr':
##
##      combine
## The following object is masked from 'package:psych':
##
##      outlier
## The following object is masked from 'package:ggplot2':
##
##      margin

```

```

library(e1071)
library(nnet)
library(CORElearn)

##
## Attaching package: 'CORElearn'

## The following object is masked from 'package:survey':
##
##      calibrate

Loading Original Data
filename = 'C:\\Users\\guohr\\Desktop\\readm_pre\\diabetic_data.csv'
filename2 = 'C:\\Users\\Mice\\Desktop\\Statistic_pj\\diabetic_data.csv'
data = read.table(filename, sep = ",", header = T, na.strings = "?")
nrow(data)

## [1] 101766

Dropping some columns
data = select(data, -encounter_id, -weight, -payer_code, -(25:41), -(43:47))

Adding missing value
any(is.na(data$race)) # true

## [1] TRUE
any(is.na(data$medical_specialty)) # true

## [1] TRUE
levels <- levels(data$race)
levels[length(levels) + 1] <- "Missing"

# refactor Species to include "Missing" as a factor level
# and replace NA with "None"
data$race <- factor(data$race, levels = levels)
data$race[is.na(data$race)] <- "Missing"

levels <- levels(data$medical_specialty)
levels[length(levels) + 1] <- "Missing"

data$medical_specialty <- factor(data$medical_specialty, levels = levels)
data$medical_specialty[is.na(data$medical_specialty)] <- "Missing"

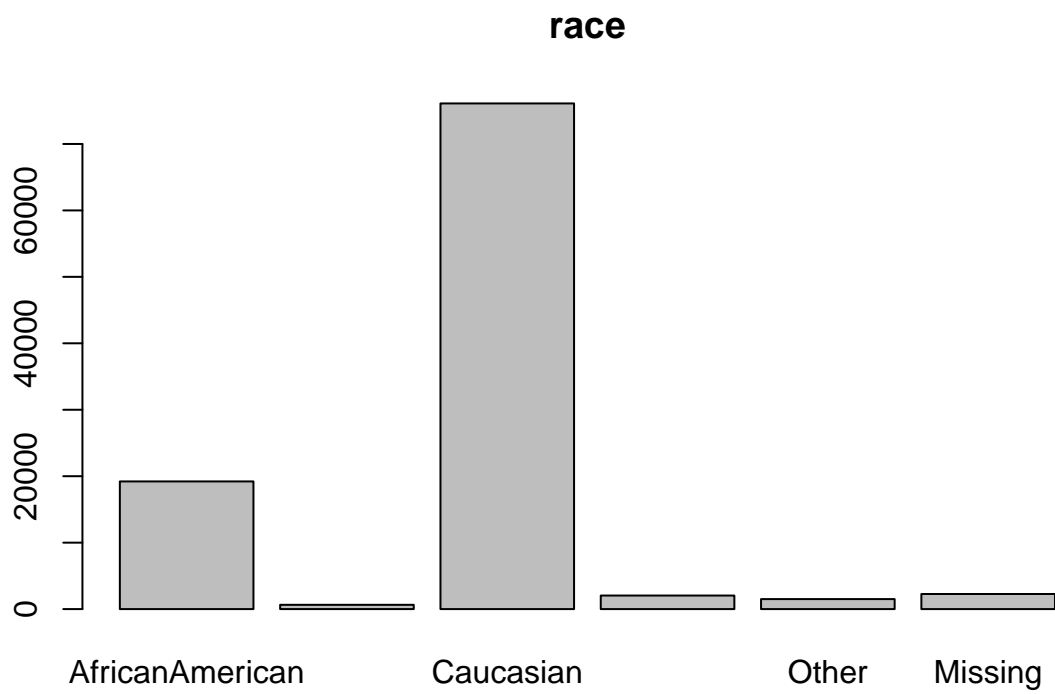
any(is.na(data$race)) # false

## [1] FALSE
any(is.na(data$medical_specialty)) # false

## [1] FALSE

Preliminary data analysis
plot(data$race, main = "race")

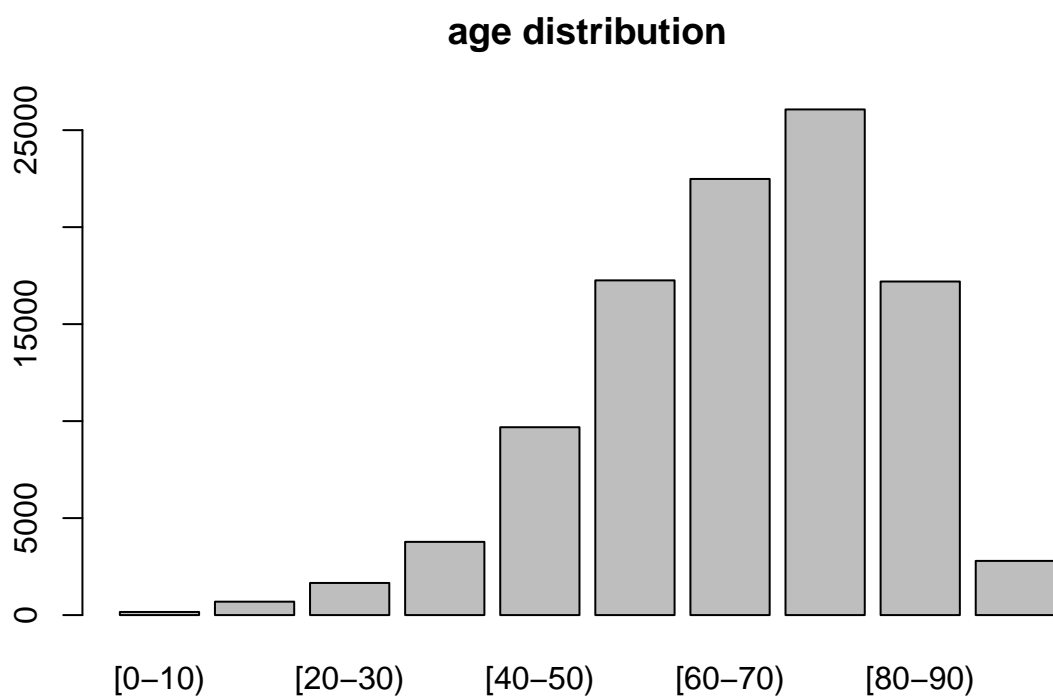
```



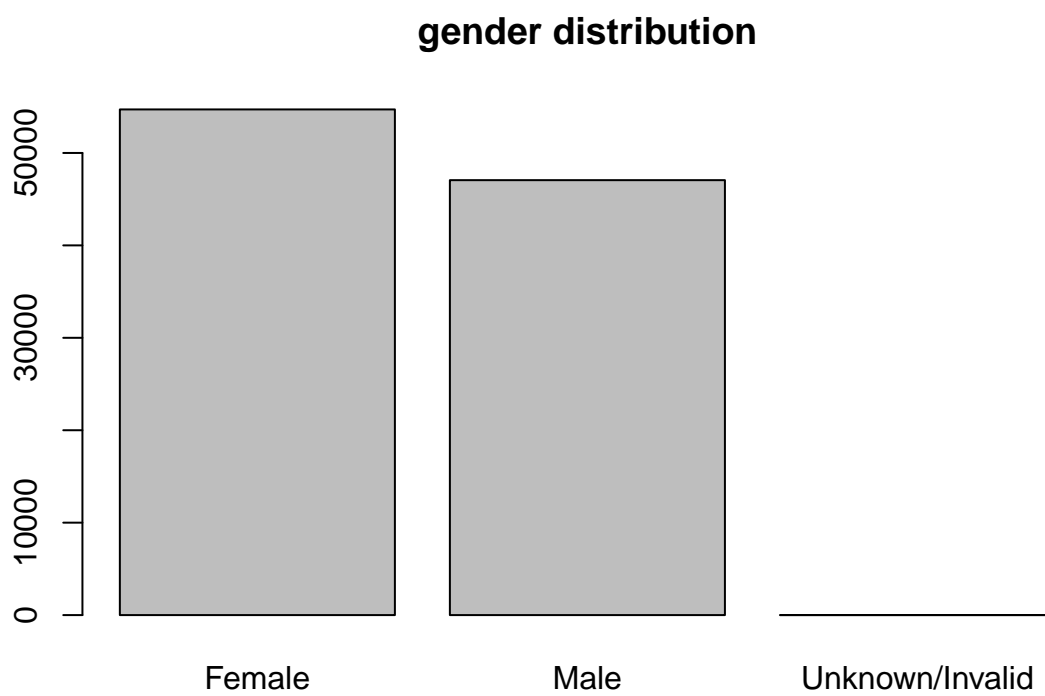
```
# PLOTS
```

```
# variable distributions
```

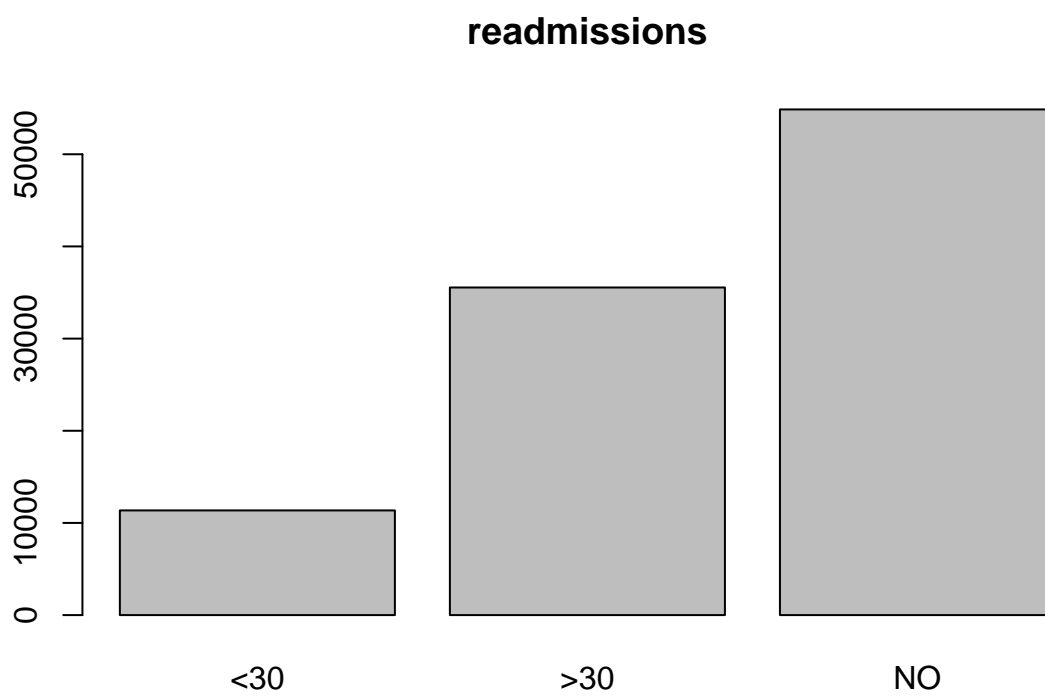
```
plot(data$age, main = "age distribution")
```



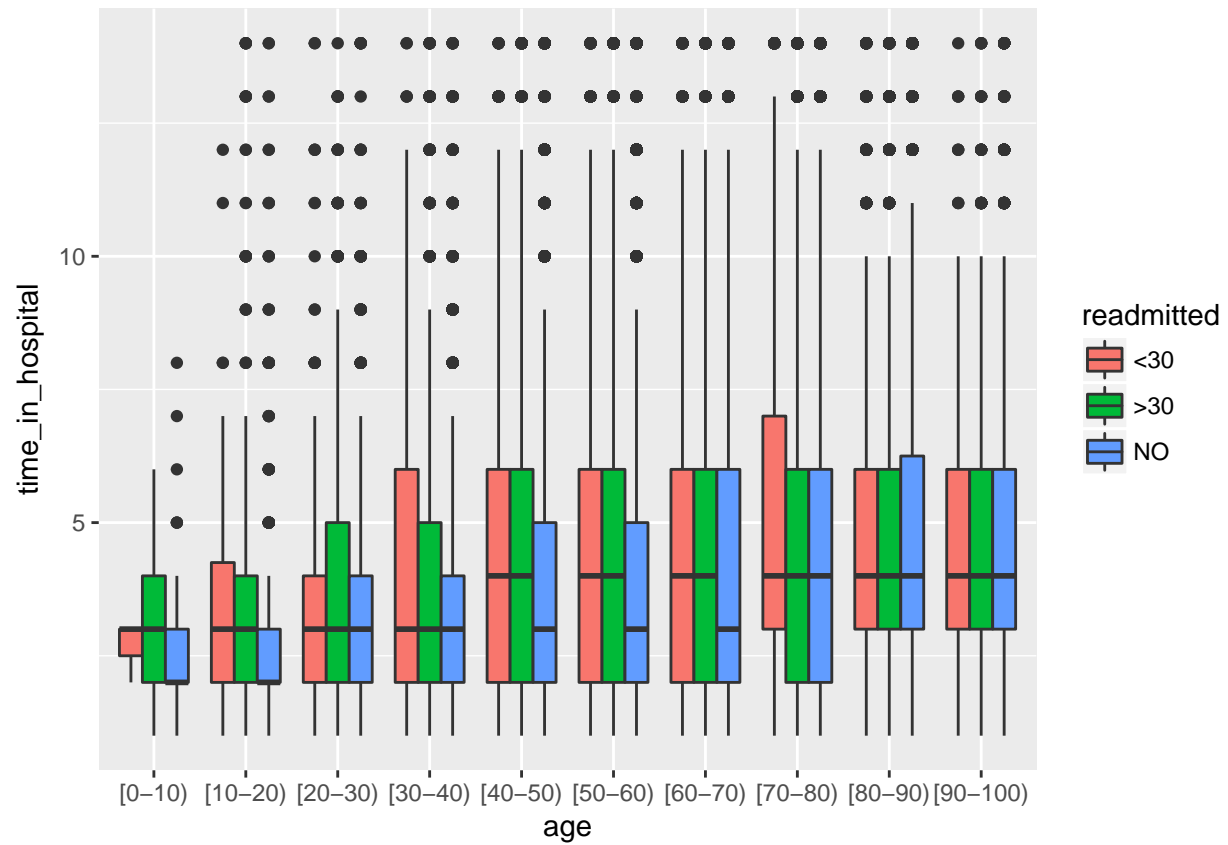
```
plot(data$gender, main = "gender distribution")
```



```
plot(data$readmitted, main = "readmissions")
```

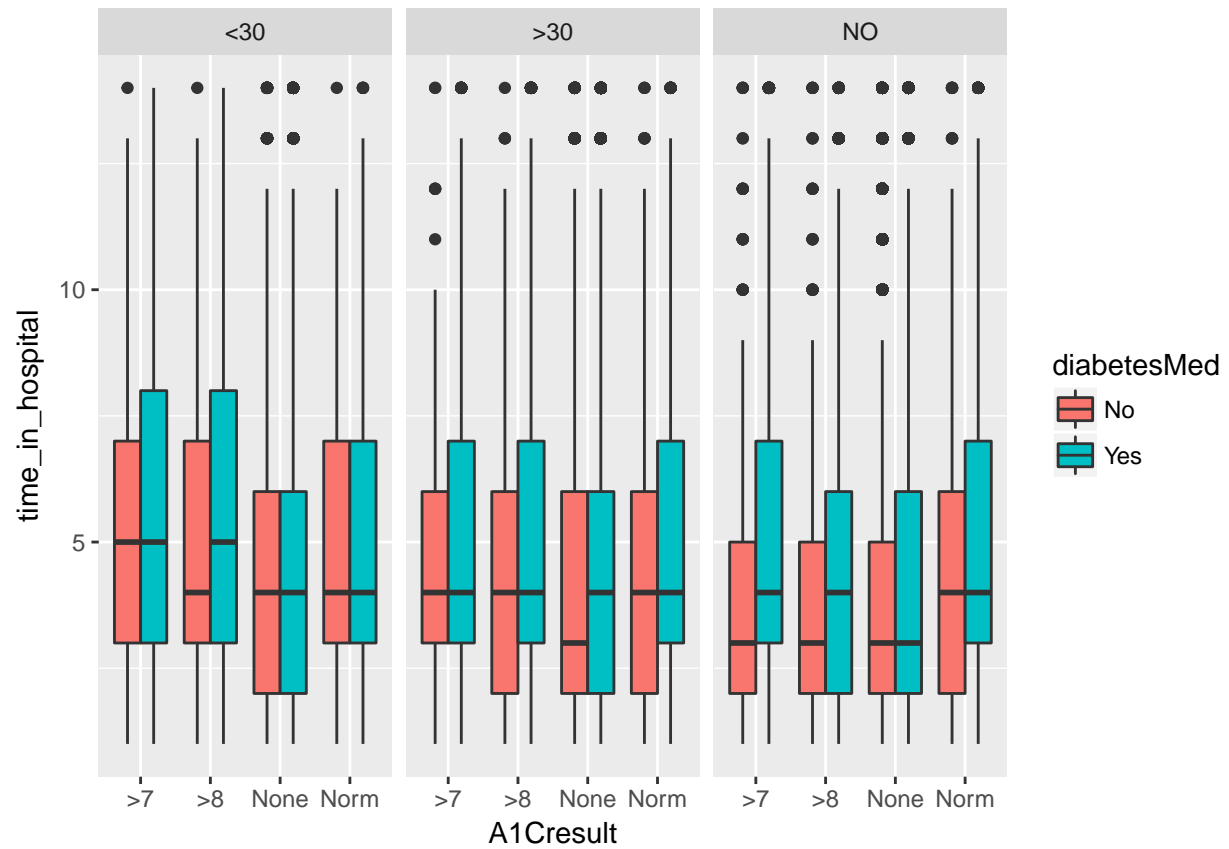


```
g <- ggplot(data, aes(x=age, y=time_in_hospital))  
g + geom_boxplot(aes(fill=readmitted))
```

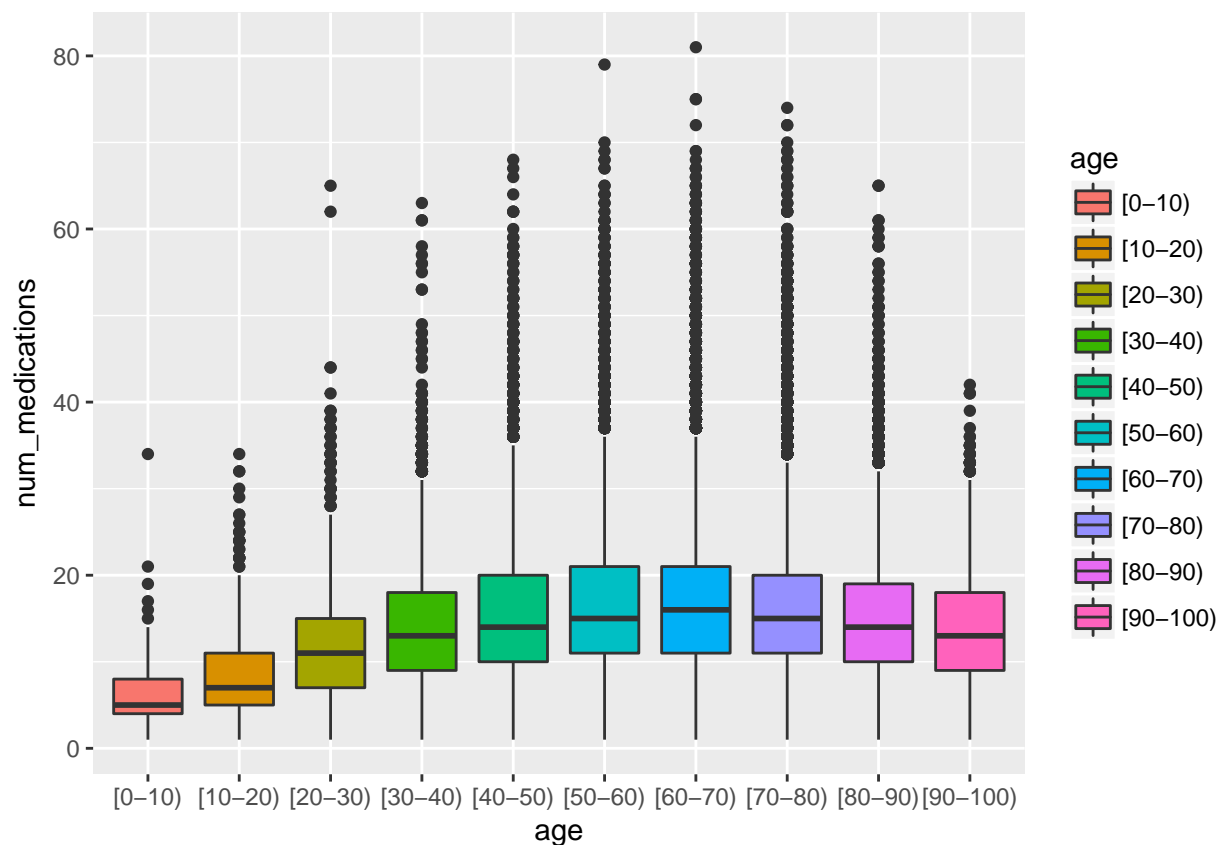


```
g <- ggplot(data,aes(x=A1Cresult, y=time_in_hospital))
g + geom_boxplot(aes(fill=diabetesMed)) + facet_grid(. ~ readmitted)
```





```
g <- ggplot(data,aes(x=age, y=num_medications))
g + geom_boxplot(aes(fill=age))
```

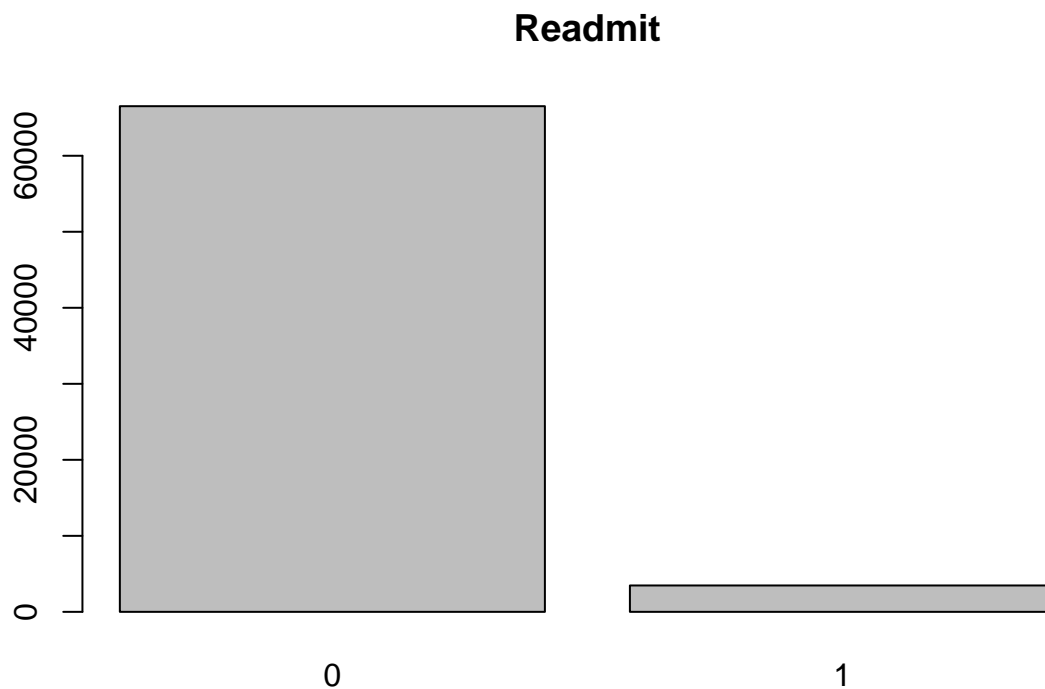


Preprocessing data with RSQL, merging classes of A1Cresult and readmitted, also filter out instances with discharge\_disposition\_id

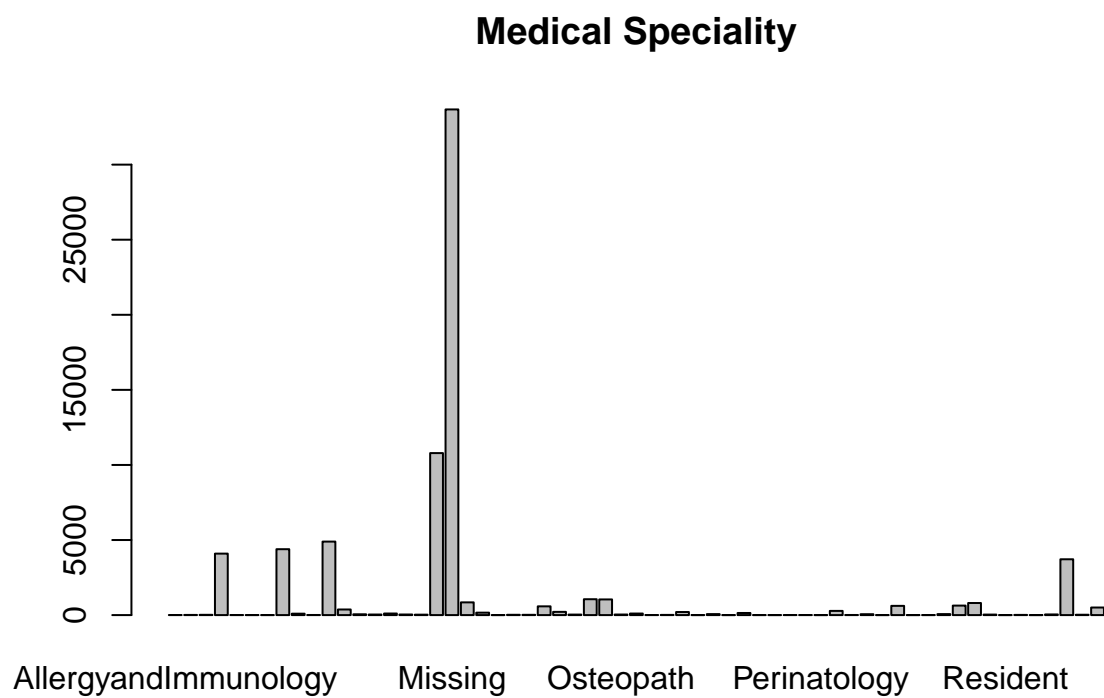
```
data = sqldf("select *, count(distinct patient_nbr) as a from data where discharge_disposition_id not in (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,81,82,83,84,85,86,87,88,89,90,91,92,93,94,95,96,97,98,99,100,101,102,103,104,105,106,107,108,109,110,111,112,113,114,115,116,117,118,119,120,121,122,123,124,125,126,127,128,129,130,131,132,133,134,135,136,137,138,139,140,141,142,143,144,145,146,147,148,149,150,151,152,153,154,155,156,157,158,159,160,161,162,163,164,165,166,167,168,169,170,171,172,173,174,175,176,177,178,179,180,181,182,183,184,185,186,187,188,189,190,191,192,193,194,195,196,197,198,199,200,201,202,203,204,205,206,207,208,209,210,211,212,213,214,215,216,217,218,219,220,221,222,223,224,225,226,227,228,229,230,231,232,233,234,235,236,237,238,239,240,241,242,243,244,245,246,247,248,249,250,251,252,253,254,255,256,257,258,259,260,261,262,263,264,265,266,267,268,269,270,271,272,273,274,275,276,277,278,279,280,281,282,283,284,285,286,287,288,289,290,291,292,293,294,295,296,297,298,299,300,301,302,303,304,305,306,307,308,309,310,311,312,313,314,315,316,317,318,319,320,321,322,323,324,325,326,327,328,329,330,331,332,333,334,335,336,337,338,339,340,341,342,343,344,345,346,347,348,349,350,351,352,353,354,355,356,357,358,359,360,361,362,363,364,365,366,367,368,369,370,371,372,373,374,375,376,377,378,379,380,381,382,383,384,385,386,387,388,389,390,391,392,393,394,395,396,397,398,399,400,401,402,403,404,405,406,407,408,409,410,411,412,413,414,415,416,417,418,419,420,421,422,423,424,425,426,427,428,429,430,431,432,433,434,435,436,437,438,439,440,441,442,443,444,445,446,447,448,449,450,451,452,453,454,455,456,457,458,459,460,461,462,463,464,465,466,467,468,469,470,471,472,473,474,475,476,477,478,479,480,481,482,483,484,485,486,487,488,489,490,491,492,493,494,495,496,497,498,499,500,501,502,503,504,505,506,507,508,509,510,511,512,513,514,515,516,517,518,519,520,521,522,523,524,525,526,527,528,529,530,531,532,533,534,535,536,537,538,539,540,541,542,543,544,545,546,547,548,549,550,551,552,553,554,555,556,557,558,559,560,561,562,563,564,565,566,567,568,569,570,571,572,573,574,575,576,577,578,579,580,581,582,583,584,585,586,587,588,589,590,591,592,593,594,595,596,597,598,599,600,601,602,603,604,605,606,607,608,609,610,611,612,613,614,615,616,617,618,619,620,621,622,623,624,625,626,627,628,629,630,631,632,633,634,635,636,637,638,639,640,641,642,643,644,645,646,647,648,649,650,651,652,653,654,655,656,657,658,659,660,661,662,663,664,665,666,667,668,669,670,671,672,673,674,675,676,677,678,679,680,681,682,683,684,685,686,687,688,689,690,691,692,693,694,695,696,697,698,699,700,701,702,703,704,705,706,707,708,709,710,711,712,713,714,715,716,717,718,719,720,721,722,723,724,725,726,727,728,729,730,731,732,733,734,735,736,737,738,739,740,741,742,743,744,745,746,747,748,749,750,751,752,753,754,755,756,757,758,759,760,761,762,763,764,765,766,767,768,769,770,771,772,773,774,775,776,777,778,779,780,781,782,783,784,785,786,787,788,789,790,791,792,793,794,795,796,797,798,799,800,801,802,803,804,805,806,807,808,809,810,811,812,813,814,815,816,817,818,819,820,821,822,823,824,825,826,827,828,829,830,831,832,833,834,835,836,837,838,839,840,841,842,843,844,845,846,847,848,849,850,851,852,853,854,855,856,857,858,859,860,861,862,863,864,865,866,867,868,869,870,871,872,873,874,875,876,877,878,879,880,881,882,883,884,885,886,887,888,889,890,891,892,893,894,895,896,897,898,899,900,901,902,903,904,905,906,907,908,909,910,911,912,913,914,915,916,917,918,919,920,921,922,923,924,925,926,927,928,929,930,931,932,933,934,935,936,937,938,939,940,941,942,943,944,945,946,947,948,949,950,951,952,953,954,955,956,957,958,959,960,961,962,963,964,965,966,967,968,969,970,971,972,973,974,975,976,977,978,979,980,981,982,983,984,985,986,987,988,989,990,991,992,993,994,995,996,997,998,999,1000,1001,1002,1003,1004,1005,1006,1007,1008,1009,1010,1011,1012,1013,1014,1015,1016,1017,1018,1019,1020,1021,1022,1023,1024,1025,1026,1027,1028,1029,1030,1031,1032,1033,1034,1035,1036,1037,1038,1039,1040,1041,1042,1043,1044,1045,1046,1047,1048,1049,1050,1051,1052,1053,1054,1055,1056,1057,1058,1059,1060,1061,1062,1063,1064,1065,1066,1067,1068,1069,1070,1071,1072,1073,1074,1075,1076,1077,1078,1079,1080,1081,1082,1083,1084,1085,1086,1087,1088,1089,1090,1091,1092,1093,1094,1095,1096,1097,1098,1099,1100,1101,1102,1103,1104,1105,1106,1107,1108,1109,1110,1111,1112,1113,1114,1115,1116,1117,1118,1119,1120,1121,1122,1123,1124,1125,1126,1127,1128,1129,1130,1131,1132,1133,1134,1135,1136,1137,1138,1139,1140,1141,1142,1143,1144,1145,1146,1147,1148,1149,1150,1151,1152,1153,1154,1155,1156,1157,1158,1159,1160,1161,1162,1163,1164,1165,1166,1167,1168,1169,1170,1171,1172,1173,1174,1175,1176,1177,1178,1179,1180,1181,1182,1183,1184,1185,1186,1187,1188,1189,1190,1191,1192,1193,1194,1195,1196,1197,1198,1199,1200,1201,1202,1203,1204,1205,1206,1207,1208,1209,1210,1211,1212,1213,1214,1215,1216,1217,1218,1219,1220,1221,1222,1223,1224,1225,1226,1227,1228,1229,1230,1231,1232,1233,1234,1235,1236,1237,1238,1239,1240,1241,1242,1243,1244,1245,1246,1247,1248,1249,1250,1251,1252,1253,1254,1255,1256,1257,1258,1259,1260,1261,1262,1263,1264,1265,1266,1267,1268,1269,1270,1271,1272,1273,1274,1275,1276,1277,1278,1279,1280,1281,1282,1283,1284,1285,1286,1287,1288,1289,1290,1291,1292,1293,1294,1295,1296,1297,1298,1299,1300,1301,1302,1303,1304,1305,1306,1307,1308,1309,1310,1311,1312,1313,1314,1315,1316,1317,1318,1319,1320,1321,1322,1323,1324,1325,1326,1327,1328,1329,1330,1331,1332,1333,1334,1335,1336,1337,1338,1339,1340,1341,1342,1343,1344,1345,1346,1347,1348,1349,1350,1351,1352,1353,1354,1355,1356,1357,1358,1359,1360,1361,1362,1363,1364,1365,1366,1367,1368,1369,1370,1371,1372,1373,1374,1375,1376,1377,1378,1379,1380,1381,1382,1383,1384,1385,1386,1387,1388,1389,1390,1391,1392,1393,1394,1395,1396,1397,1398,1399,1400,1401,1402,1403,1404,1405,1406,1407,1408,1409,1410,1411,1412,1413,1414,1415,1416,1417,1418,1419,1420,1421,1422,1423,1424,1425,1426,1427,1428,1429,1430,1431,1432,1433,1434,1435,1436,1437,1438,1439,1440,1441,1442,1443,1444,1445,1446,1447,1448,1449,1450,1451,1452,1453,1454,1455,1456,1457,1458,1459,1460,1461,1462,1463,1464,1465,1466,1467,1468,1469,1470,1471,1472,1473,1474,1475,1476,1477,1478,1479,1480,1481,1482,1483,1484,1485,1486,1487,1488,1489,1490,1491,1492,1493,1494,1495,1496,1497,1498,1499,1500,1501,1502,1503,1504,1505,1506,1507,1508,1509,1510,1511,1512,1513,1514,1515,1516,1517,1518,1519,1520,1521,1522,1523,1524,1525,1526,1527,1528,1529,1530,1531,1532,1533,1534,1535,1536,1537,1538,1539,1540,1541,1542,1543,1544,1545,1546,1547,1548,1549,1550,1551,1552,1553,1554,1555,1556,1557,1558,1559,1560,1561,1562,1563,1564,1565,1566,1567,1568,1569,1570,1571,1572,1573,1574,1575,1576,1577,1578,1579,1580,1581,1582,1583,1584,1585,1586,1587,1588,1589,1590,1591,1592,1593,1594,1595,1596,1597,1598,1599,1600,1601,1602,1603,1604,1605,1606,1607,1608,1609,1610,1611,1612,1613,1614,1615,1616,1617,1618,1619,1620,1621,1622,1623,1624,1625,1626,1627,1628,1629,1630,1631,1632,1633,1634,1635,1636,1637,1638,1639,1640,1641,1642,1643,1644,1645,1646,1647,1648,1649,1650,1651,1652,1653,1654,1655,1656,1657,1658,1659,1660,1661,1662,1663,1664,1665,1666,1667,1668,1669,1670,1671,1672,1673,1674,1675,1676,1677,1678,1679,1680,1681,1682,1683,1684,1685,1686,1687,1688,1689,1690,1691,1692,1693,1694,1695,1696,1697,1698,1699,1700,1701,1702,1703,1704,1705,1706,1707,1708,1709,1710,1711,1712,1713,1714,1715,1716,1717,1718,1719,1720,1721,1722,1723,1724,1725,1726,1727,1728,1729,1730,1731,1732,1733,1734,1735,1736,1737,1738,1739,1740,1741,1742,1743,1744,1745,1746,1747,1748,1749,1750,1751,1752,1753,1754,1755,1756,1757,1758,1759,1760,1761,1762,1763,1764,1765,1766,1767,1768,1769,1770,1771,1772,1773,1774,1775,1776,1777,1778,1779,1780,1781,1782,1783,1784,1785,1786,1787,1788,1789,1790,1791,1792,1793,1794,1795,1796,1797,1798,1799,1800,1801,1802,1803,1804,1805,1806,1807,1808,1809,1810,1811,1812,1813,1814,1815,1816,1817,1818,1819,1820,1821,1822,1823,1824,1825,1826,1827,1828,1829,1830,1831,1832,1833,1834,1835,1836,1837,1838,1839,1840,1841,1842,1843,1844,1845,1846,1847,1848,1849,1850,1851,1852,1853,1854,1855,1856,1857,1858,1859,1860,1861,1862,1863,1864,1865,1866,1867,1868,1869,1870,1871,1872,1873,1874,1875,1876,1877,1878,1879,1880,1881,1882,1883,1884,1885,1886,1887,1888,1889,1890,1891,1892,1893,1894,1895,1896,1897,1898,1899,1900,1901,1902,1903,1904,1905,1906,1907,1908,1909,1910,1911,1912,1913,1914,1915,1916,1917,1918,1919,1920,1921,1922,1923,1924,1925,1926,1927,1928,1929,1930,1931,1932,1933,1934,1935,1936,1937,1938,1939,1940,1941,1942,1943,1944,1945,1946,1947,1948,1949,1950,1951,1952,1953,1954,1955,1956,1957,1958,1959,1960,1961,1962,1963,1964,1965,1966,1967,1968,1969,1970,1971,1972,1973,1974,1975,1976,1977,1978,1979,1980,1981,1982,1983,1984,1985,1986,1987,1988,1989,1990,1991,1992,1993,1994,1995,1996,1997,1998,1999,2000,2001,2002,2003,2004,2005,2006,2007,2008,2009,2010,2011,2012,2013,2014,2015,2016,2017,2018,2019,2020,2021,2022,2023,2024,2025,2026,2027,2028,2029,2030,2031,2032,2033,2034,2035,2036,2037,2038,2039,2040,2041,2042,2043,2044,2045,2046,2047,2048,2049,2050,2051,2052,2053,2054,2055,2056,2057,2058,2059,2060,2061,2062,2063,2064,2065,2066,2067,2068,2069,2070,2071,2072,2073,2074,2075,2076,2077,2078,2079,2080,2081,2082,2083,2084,2085,2086,2087,2088,2089,2090,2091,2092,2093,2094,2095,2096,2097,2098,2099,2100,2101,2102,2103,2104,2105,2106,2107,2108,2109,2110,2111,2112,2113,2114,2115,2116,2117,2118,2119,2120,2121,2122,2123,2124,2125,2126,2127,2128,2129,2130,2131,2132,2133,2134,2135,2136,2137,2138,2139,2140,2141,2142,2143,2144,2145,2146,2147,2148,2149,2150,2151,2152,2153,2154,2155,2156,2157,2158,2159,2160,2161,2162,2163,2164,2165,2166,2167,2168,2169,2170,2171,2172,2173,2174,2175,2176,2177,2178,2179,2180,2181,2182,2183,2184,2185,2186,2187,2188,2189,2190,2191,2192,2193,2194,2195,2196,2197,2198,2199,2200,2201,2202,2203,2204,2205,2206,2207,2208,2209,2210,2211,2212,2213,2214,2215,2216,2217,2218,2219,2220,2221,2222,2223,2224,2225,2226,2227,2228,2229,2230,2231,2232,2233,2234,2235,2236,2237,2238,2239,2240,2241,2242,2243,2244,2245,2246,2247,2248,2249,2250,2251,2252,2253,2254,2255,2256,2257,2258,2259,2260,2261,2262,2263,2264,2265,2266,2267,2268,2269,2270,2271,2272,2273,2274,2275,2276,2277,2278,2279,2280,2281,2282,2283,2284,2285,2286,2287,2288,2289,2290,2291,2292,2293,2294,2295,2296,2297,2298,2299,2300,2301,2302,2303,2304,2305,2306,2307,2308,2309,2310,2311,2312,2313,2314,2315,2316,2317,2318,2319,2320,2321,2322,2323,2324,2325,2326,2327,2328,2329,2330,2331,2332,2333,2334,2335,2336,2337,2338,2339,2340,2341,2342,2343,2344,2345,2346,2347,2348,2349,2350,2351,2352,2353,2354,2355,2356,2357,2358,2359,2360,2361,2362,2363,2364,2365,2366,2367,2368,2369,2370,2371,2372,2373,2374,2375,2376,2377,2378,2379,2380,2381,2382,2383,2384,2385,2386,2387,2388,2389,2390,2391,2392,2393,2394,2395,2396,2397,2398,2399,2400,2401,2402,2403,2404,2405,2406,2407,2408,2409,2410,2411,2412,2413,2414,2415,2416,2417,2418,2419,2420,2421,2422,2423,2424,2425,2426,2427,2428,2429,2430,2431,2432,2433,2434,2435,2436,2437,2438,2439,2440,2441,2442,2443,2444,2445,2446,2447,2448,2449,2450,2451,2452,2453,2454,2455,2456,2457,2458,2459,2460,2461,2462,2463,2464,2465,2466,2467,2468,2469,2470,2471,2472,2473,2474,2475,2476,2477,2478,2479,2480,2481,2482,2483,2484,2485,2486,2487,2488,2489,2490,2491,2492,2493,2494,2495,2496,2497,2498,2499,2500,2501,2502,2503,2504,2505,2506,2507,2508,2509,2510,2511,2512,2513,2514,2515,2516,2517,2518,2519,2520,2521,2522,2523,2524,2525,2526,2527,2528,2529,2530,2531,2532,2533,2534,2535,2536,2537,2538,2539,2540,2541,2542,2543,2544,2545,2546,2547,2548,2549,2550,2551,2552,2553,2554,2555,2556,2557,2558,2559,2560,2561,2562,2563,2564,2565,2566,2567,2568,2569,2570,2571,2572,2573,2574,2575,2576,2577,2578,2579,2580,2581,2582,2583,2584,2585,2586,2587,2588,2589,2590,2591,2592,2593,2594,2595,2596,2597,2598,2599,2600,2601,2602,2603,2604,2605,2606,26
```

```
# Factorize some columns in data
data$Readmit <- as.factor(data$Readmit)
data$race <- as.factor(data$race)
data$Diag1 <- as.factor(data$Diag1)
data$Diag2 <- as.factor(data$Diag2)
data$Diag3 <- as.factor(data$Diag3)
data$HbA1c <- as.factor(data$HbA1c)
data$MedicalSpeciality <- as.factor(data$MedicalSpeciality)

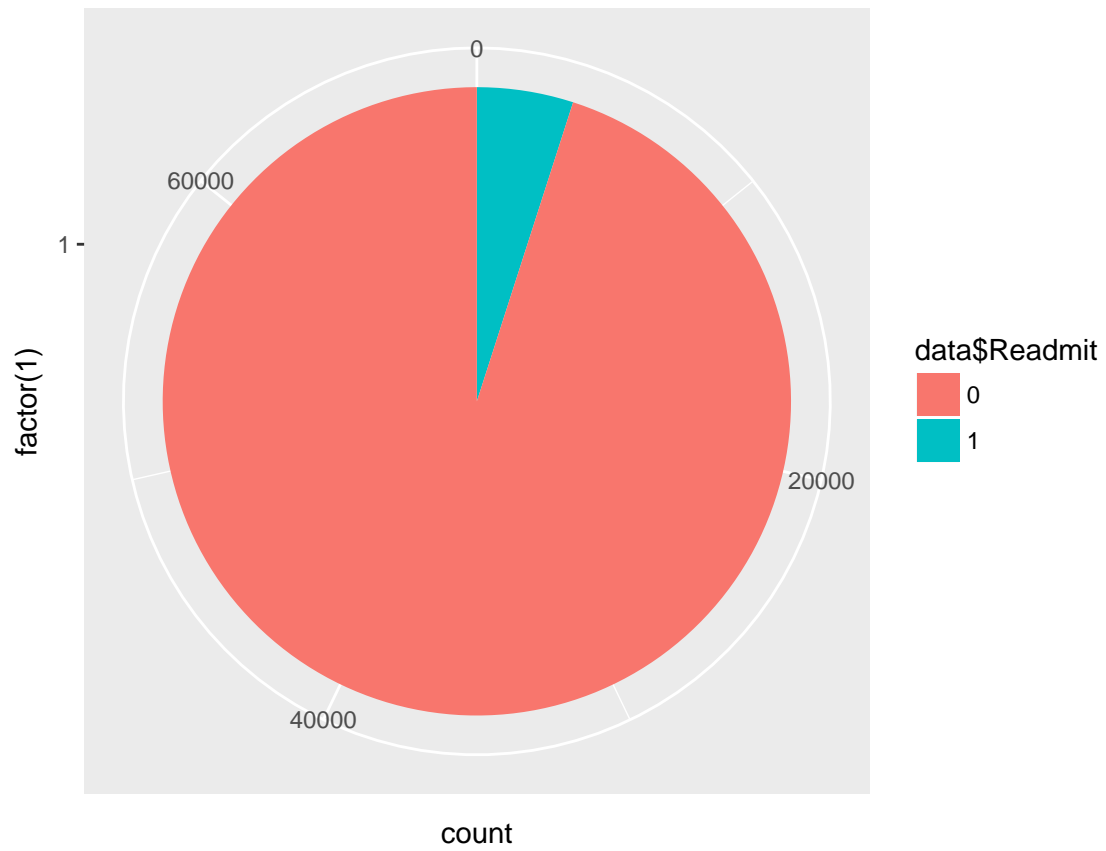
plot(data$Readmit, main="Readmit")
```



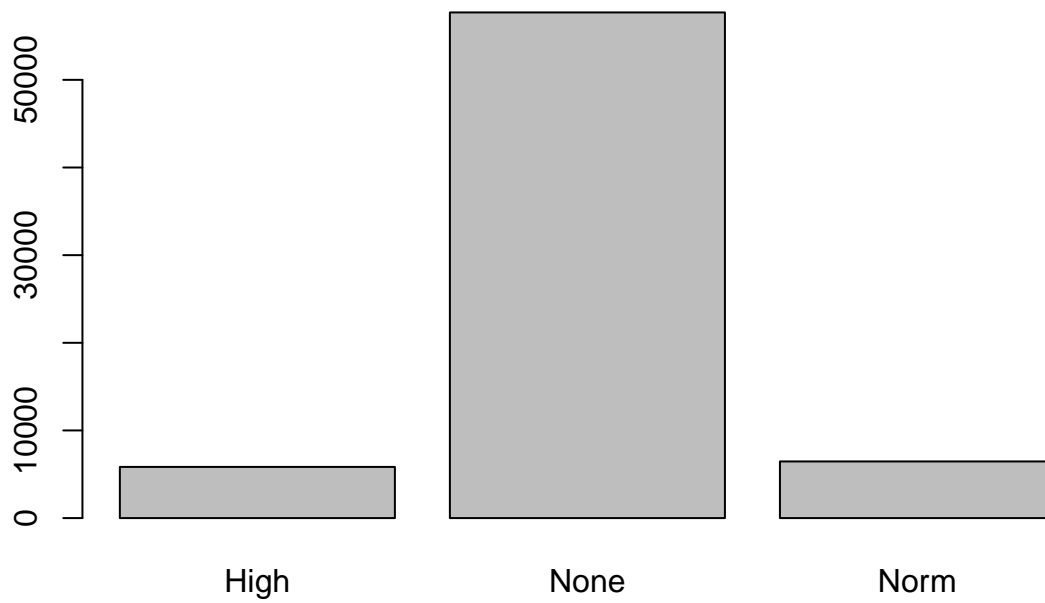
```
plot(data$MedicalSpeciality, main="Medical Speciality")
```



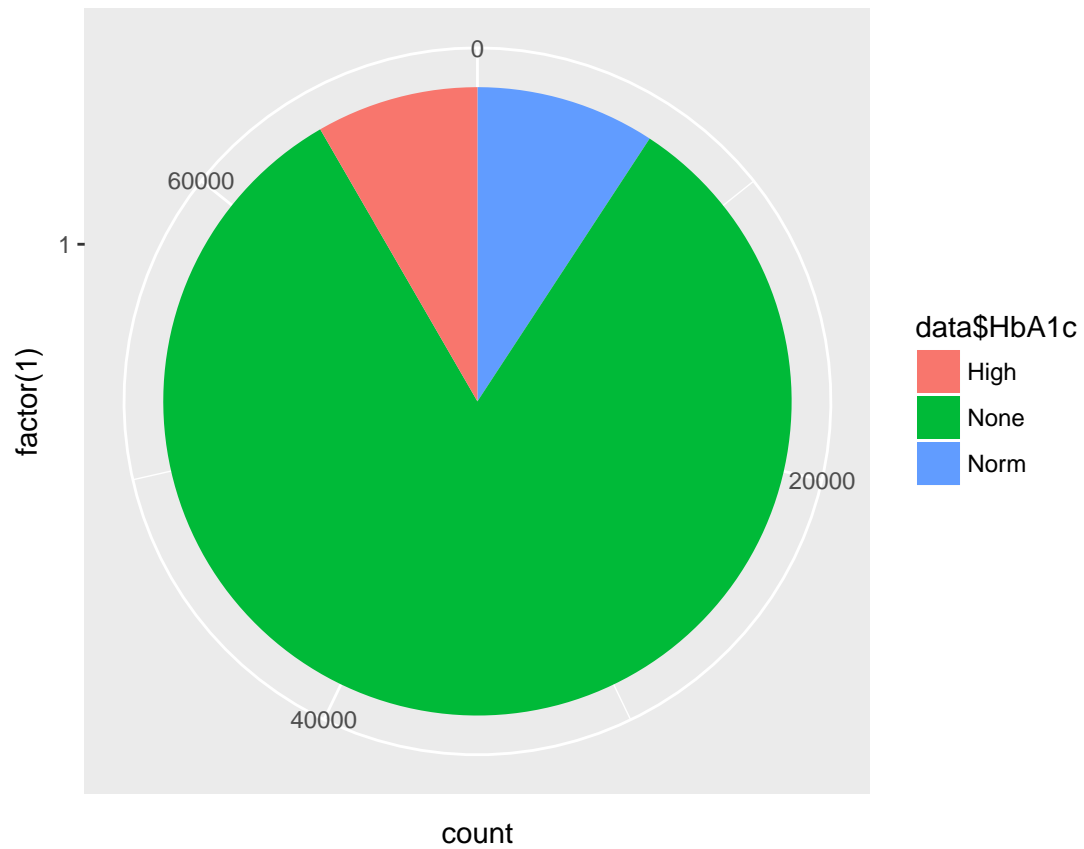
```
ggplot(data)+geom_bar(width=1, aes(x=factor(1),fill=data$Readmit))+coord_polar(theta="y")
```



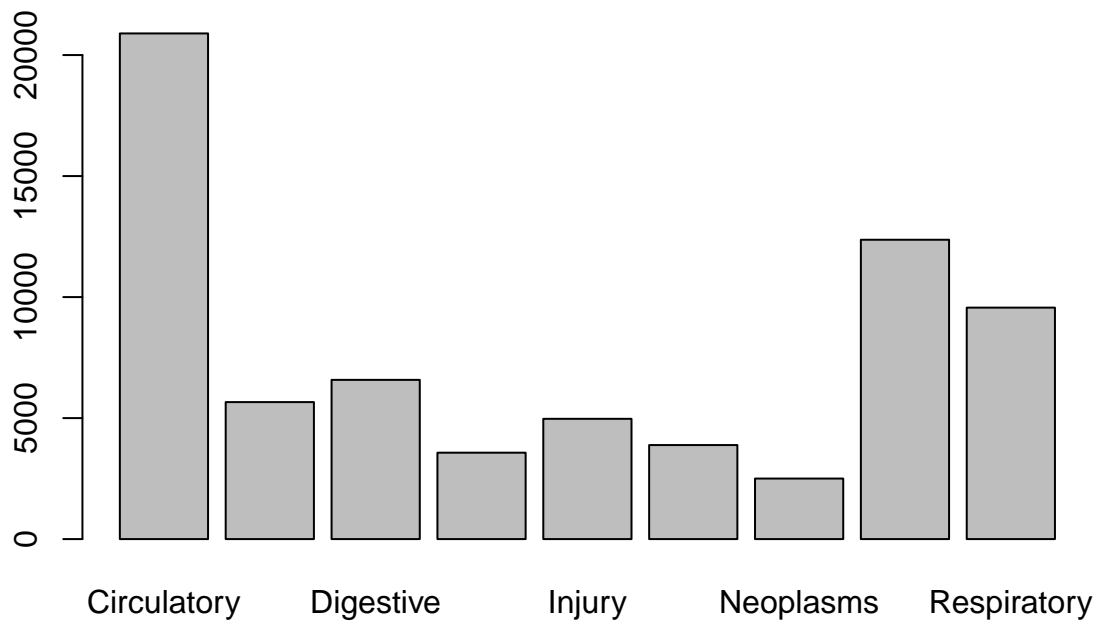
```
plot(data$HbA1c)
```



```
ggplot(data)+geom_bar(width=1, aes(x=factor(1), fill=data$HbA1c))+coord_polar(theta="y")
```

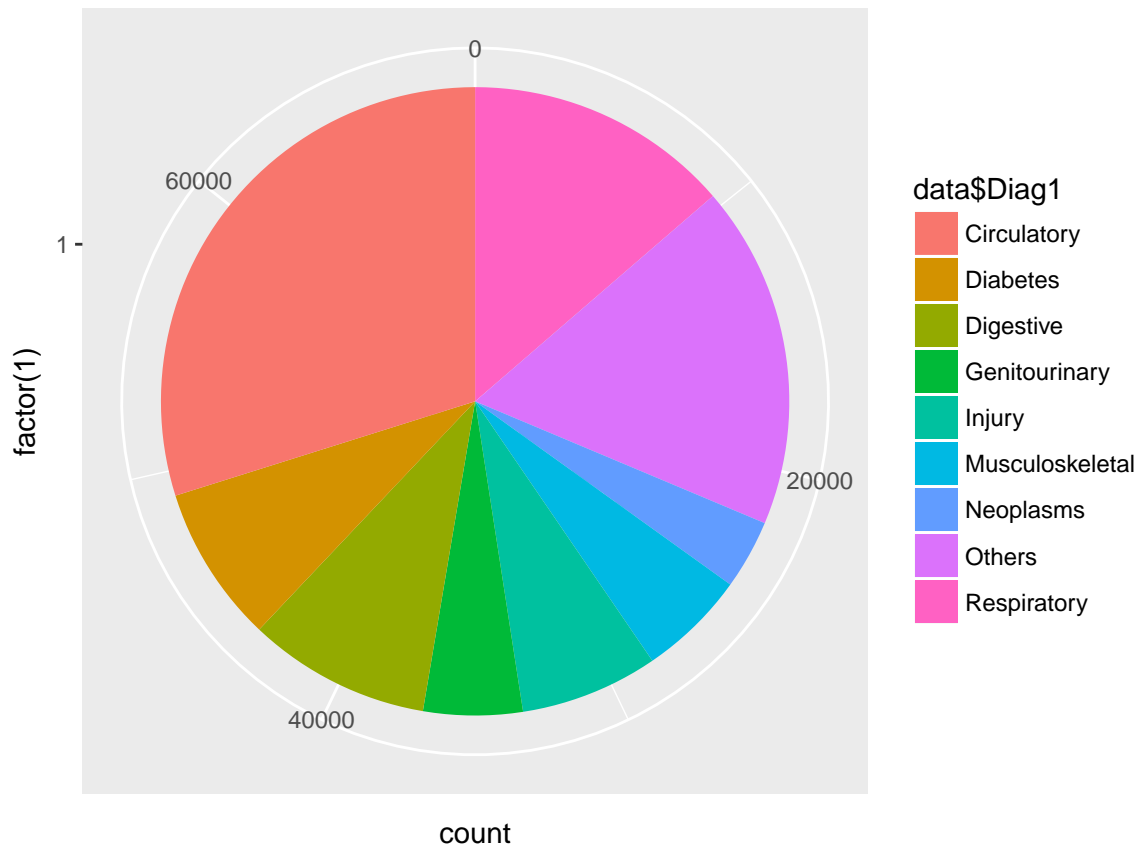


```
plot(data$Diag1)
```

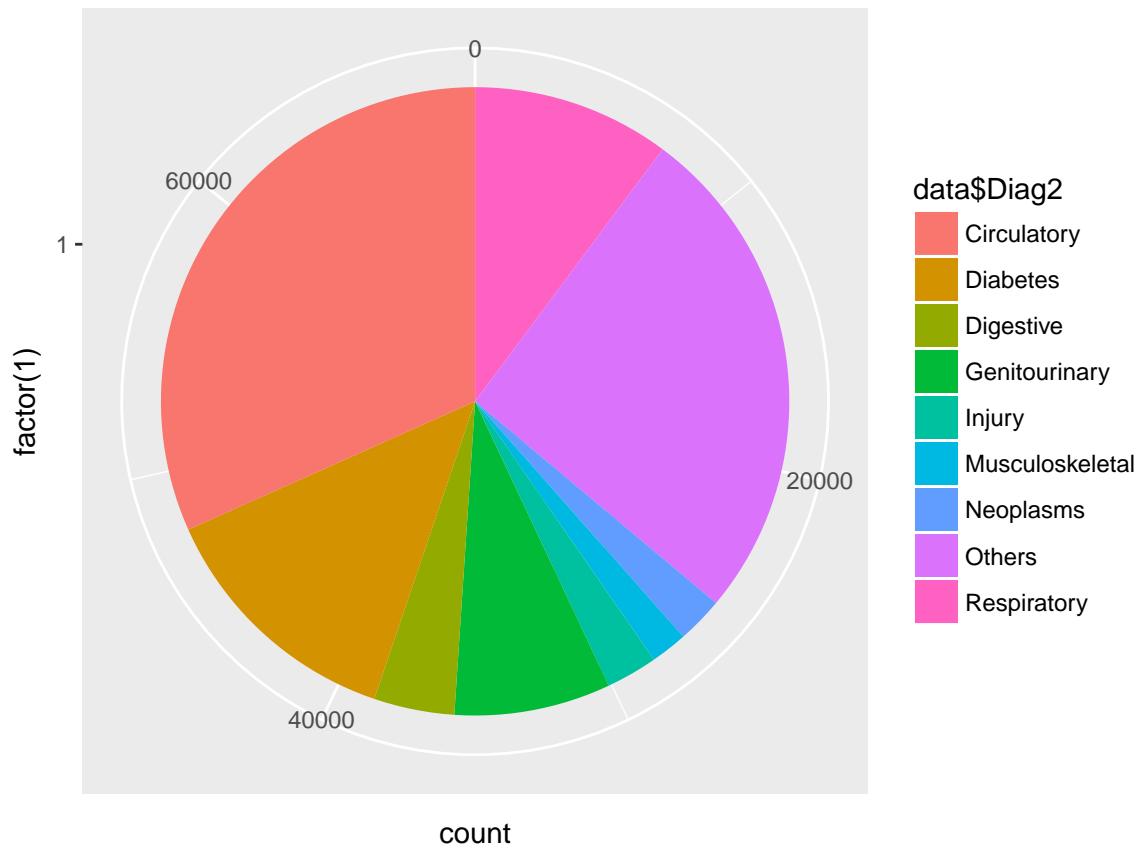


```
ggplot(data)+geom_bar(width=1, aes(x=factor(1),fill=data$Diag1))+coord_polar(theta="y")
```

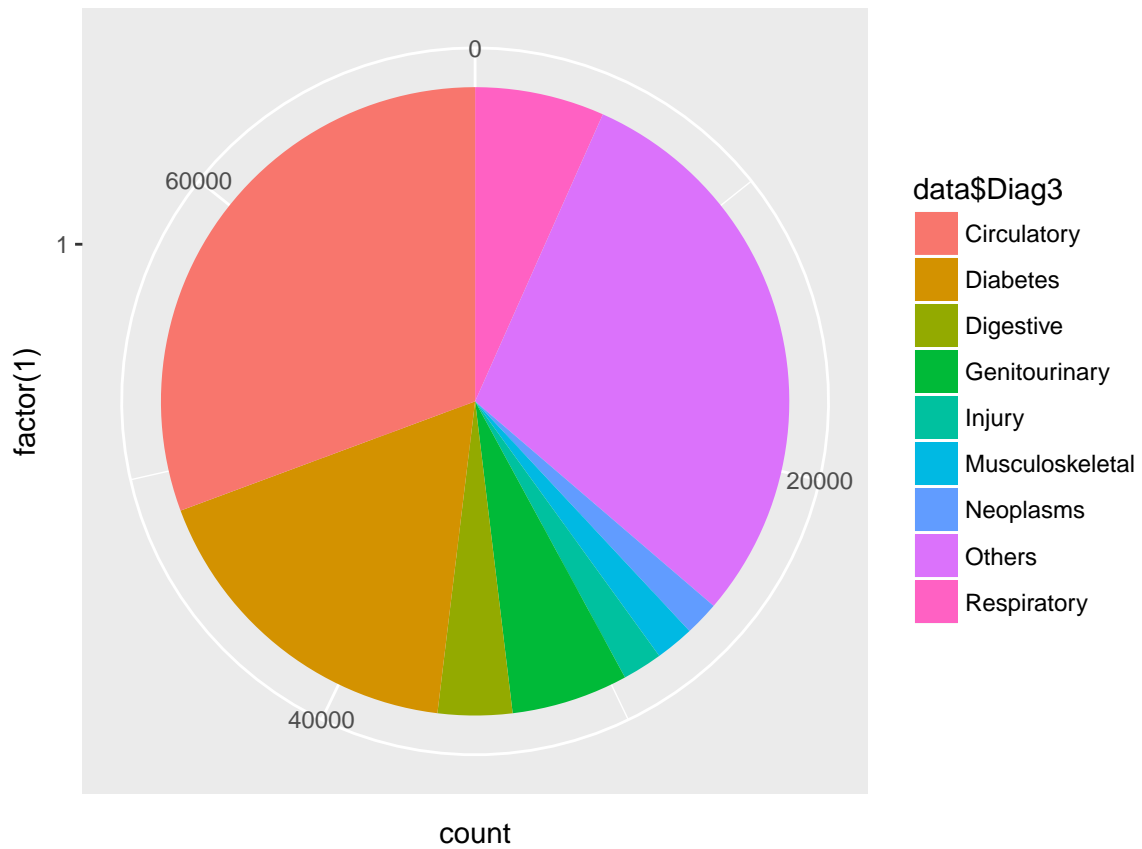




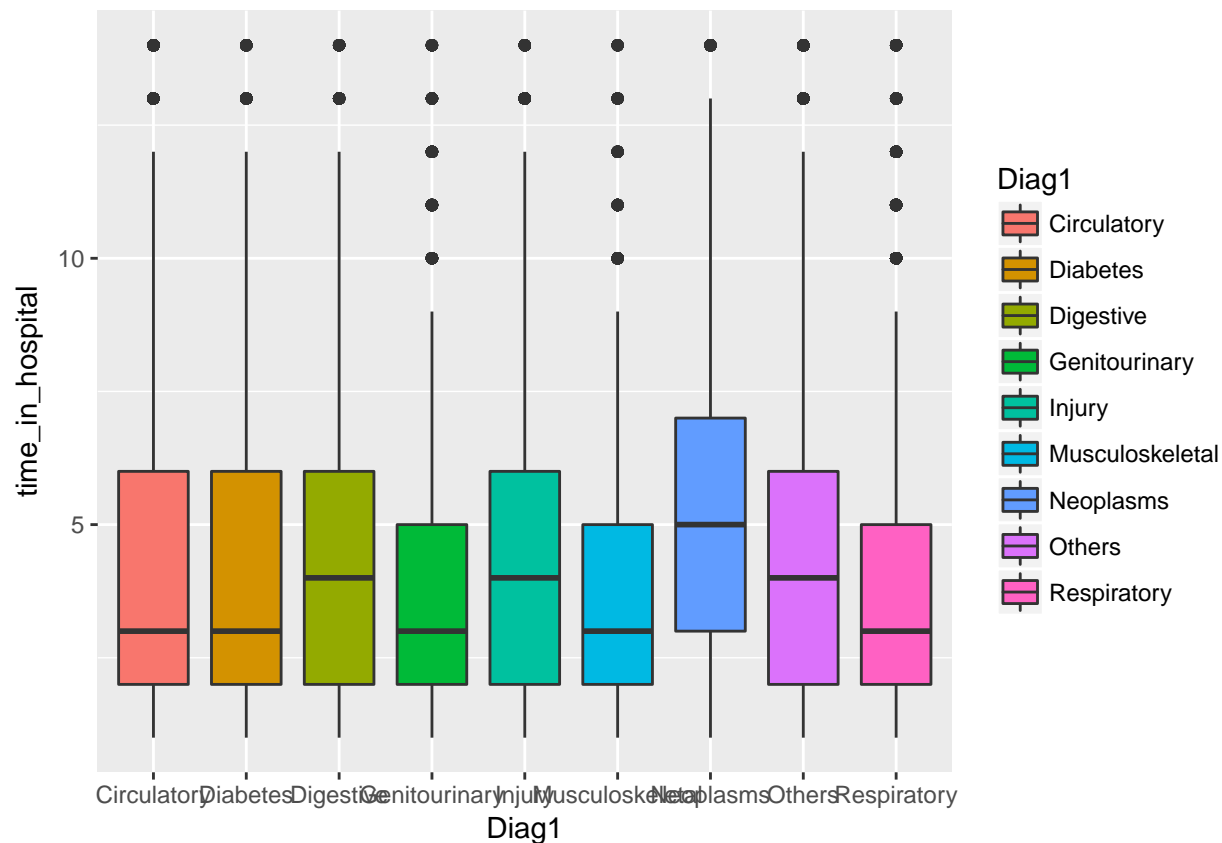
```
ggplot(data)+geom_bar(width=1, aes(x=factor(1),fill=data$Diag2))+coord_polar(theta="y")
```



```
ggplot(data)+geom_bar(width=1, aes(x=factor(1),fill=data$Diag3))+coord_polar(theta="y")
```



```
g <- ggplot(data, aes(x=Diag1, y=time_in_hospital))
g + geom_boxplot(aes(fill=Diag1))
```



Further data cleaning, remove some columns, merge column HbA1c & change to 'Reaction'

```
data = select(data, -diag_1, -diag_2, -diag_3, -admission_source_id, -num_procedures, -num_medications,
```

```
data = select(data, -Diag2, -Diag3, -number_diagnoses, -max_glu_serum, -insulin, -diabetesMed)
```

```
data = sqldf("select case when HbA1c is 'High' and change is 'Ch' then 'High&Ch' when HbA1c is 'High' and
```

```
# Factorize
```

```
data$Reaction <- as.factor(data$Reaction)
```

```
data$race <- as.factor(data$race)
```

```
data$discharge_disposition_id <- as.factor(data$discharge_disposition_id)
```

```
data$admission_type_id <- as.factor(data$admission_type_id)
```

```
# Remove duplicates
```

```
data = select(data, -Diag1, -HbA1c, -change)
```

Correlation of numeric factors

```
#data[is.na(data)] <- 0
```

```
#summary(data)
```

```
#data[] <- lapply(data, function(x) {
```

```
#   if(is.factor(x)) as.numeric(as.character(x)) else x
```

```
#})
```

```
#sapply(data, class)
```

```
#c <- cor(data[], use= "pairwise.complete.obs")
```

```
#corrplot(c)

# p = ggpairs(data_simp)
# p_ <- GGally::print_if_interactive
# data_simp[] <- lapply(data_simp, function(x) {
#   if(is.factor(x)) as.numeric(as.character(x)) else x
# })
# sapply(data_simp, class)
# p_(p)

# Bad result!
```

Quick PCA with numeric variables

```
# y <- select(data, Readmit)
# X <- select(data, time_in_hospital, num_procedures,
#   number_outpatient, number_emergency, number_inpatient, number_diagnoses,
#   max_glu_serum, insulin, change)
# # no rotation
# pca_noRot <- principal(X, nfactors = 5, rotate = "none")
# rotation2_noRot <- data.frame(cbind(pca_noRot$score, y))
# head(rotation2_noRot)
# pca_noRot$loadings
```

pc1 number of medications and time in hospital pc2 number of in-patient visits and emergency pc3 number of procedures pc4 number of out-patient visits pc5 number of diagnoses

Logit regression

```
logisticPseudoR2s <- function(LogModel) {
  dev <- LogModel$deviance
  nullDev <- LogModel$null.deviance
  modelN <- length(LogModel$fitted.values)
  R.l <- 1 - dev / nullDev
  R.cs <- 1- exp ( -(nullDev - dev) / modelN)
  R.n <- R.cs / ( 1 - ( exp (-(nullDev / modelN))))
  cat("Pseudo R^2 for logistic regression\n")
  cat("Hosmer and Lemeshow R^2 ", round(R.l, 3), "\n")
  cat("Cox and Snell R^2      ", round(R.cs, 3), "\n")
  cat("Nagelkerke R^2         ", round(R.n, 3), "\n")
}
```

Chisq test, select important factors (Gender could be dropped)

```
#anova(linModel_noRot, test="Chisq")
#
#
#           Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
# NULL                                69989      27608
# Reaction              3    10.73    69986      27598 0.0132927 *
# Pri_diag              8    31.23    69978      27566 0.0001279 ***
# MedicalSpeciality     58   244.94    69920      27321 < 2.2e-16 ***
# race                  5    88.91    69915      27233 < 2.2e-16 ***
# gender                2     0.69    69913      27232 0.7099052
# age                   9   160.53    69904      27071 < 2.2e-16 ***
# admission_type_id      7   174.54    69897      26897 < 2.2e-16 ***
# discharge_disposition_id 20  701.57    69877      26195 < 2.2e-16 ***
```

```
# time_in_hospital      1      21.16      69876      26174 4.218e-06 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Data Merging - Select the attributes and groups we use to train the model.

```
#summary(data2)
data2 = data
#library(sqldf)
data2$race<- as.factor(data2$race)
data2 = select(data2, -gender)
data2 = sqldf("select case discharge_disposition_id when 1 then 'Home' else 'Other' end as Discharge, *
data2 = sqldf("select case admission_type_id when 1 or 2 then 'Emergency' when 7 then 'referral' else '
data2 = sqldf("select case race when 'AfricanAmerican' then 'AfricanAmerican' when 'Caucasian' then 'Ca
data2 = sqldf("select case age when '[30-40)' then '[30, 60)' when '[40-50)' then '[30, 60)' when '[50-
data2 = sqldf("select case MedicalSpeciality when 'Missing' then 'Missing' when 'Cardiology' then 'Card
data2$race_<-as.factor(data2$race_)
data2$Admission<-as.factor(data2$Admission)
data2$Discharge<-as.factor(data2$Discharge)
data2$Pri_diag <- as.factor(data2$Pri_diag)
data2$Age_ <- as.factor(data2$Age_)
data2$Medical_speciality <- as.factor(data2$Medical_speciality)
data2 = select(data2, -race, -admission_type_id, -discharge_disposition_id, -age, -MedicalSpeciality)
summary(data2)
```

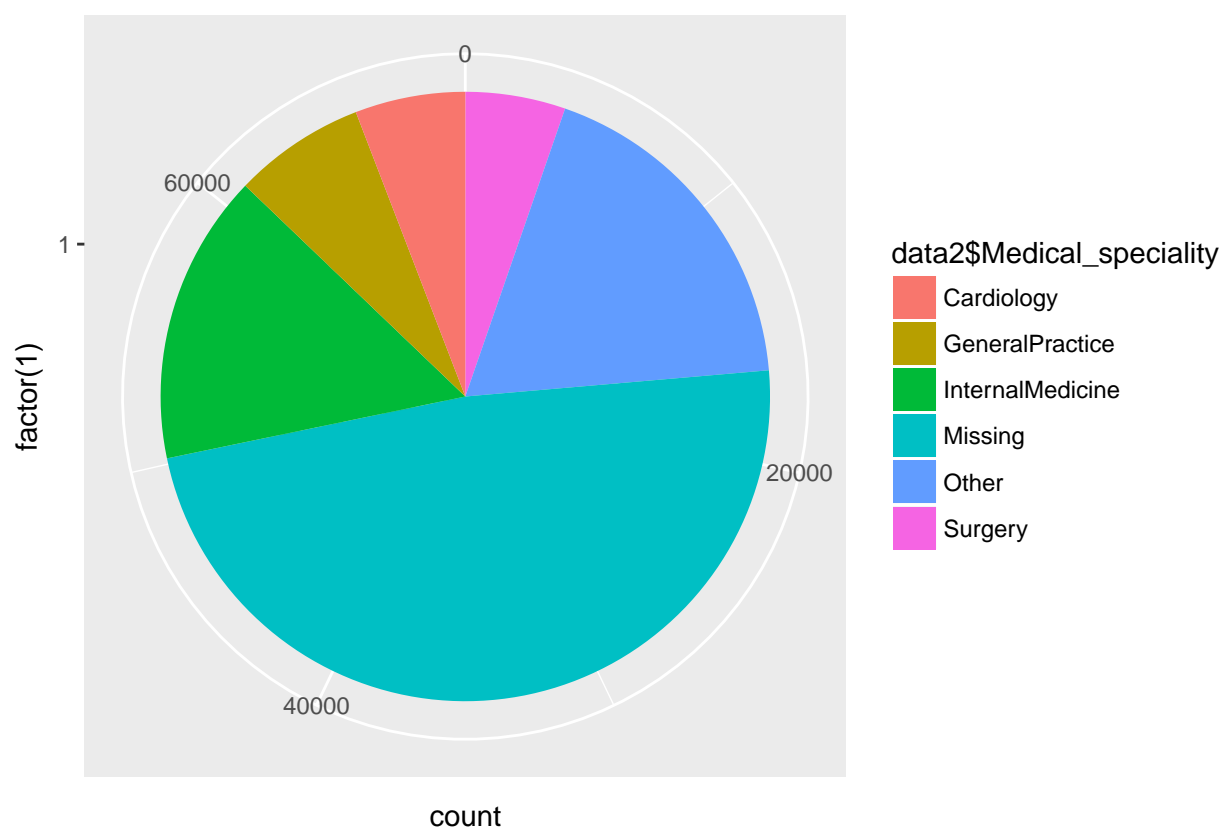
```
##           Medical_speciality           Age_           race_
## Cardiology      : 4094      [30, 60) :21587      AfricanAmerican:12656
## GeneralPractice : 4894      [60, 100):47727      Caucasian      :52352
## InternalMedicine:10788      <30      : 676      Missing      : 1850
## Missing         :33676                                     Other      : 3132
## Other           :12821
## Surgery         : 3717
##
##           Admission      Discharge      Reaction      Pri_diag
## Emergency:36098      Home :43721      High&Ch : 3784      Circulatory:20894
## Other      :33876      Other:26269      High&Not: 2053      Others      :12368
## referral : 16                                     None      :57691      Respiratory: 9564
##                                     Norm      : 6462      Digestive  : 6579
##                                     Diabetes  : 5660
##                                     Injury     : 4969
##                                     (Other)    : 9956
##
## Readmit      time_in_hospital
## 0:66521      Min.      : 1.000
## 1: 3469      1st Qu.: 2.000
##                                     Median : 4.000
##                                     Mean    : 4.302
##                                     3rd Qu.: 6.000
##                                     Max.    :14.000
##
```

```
#           Medical_speciality           Age_           race_           Admission      Discharge
# Cardiology      : 4094      [30, 60) :21587      AfricanAmerican:12656      Emergency:36098      Home :43721
# GeneralPractice : 4894      [60, 100):47727      Caucasian      :52352      Other      :33876      Other:26269
# InternalMedicine:10788      <30      : 676      Missing      : 1850      referral : 16
# Missing         :33676                                     Other      : 3132
```

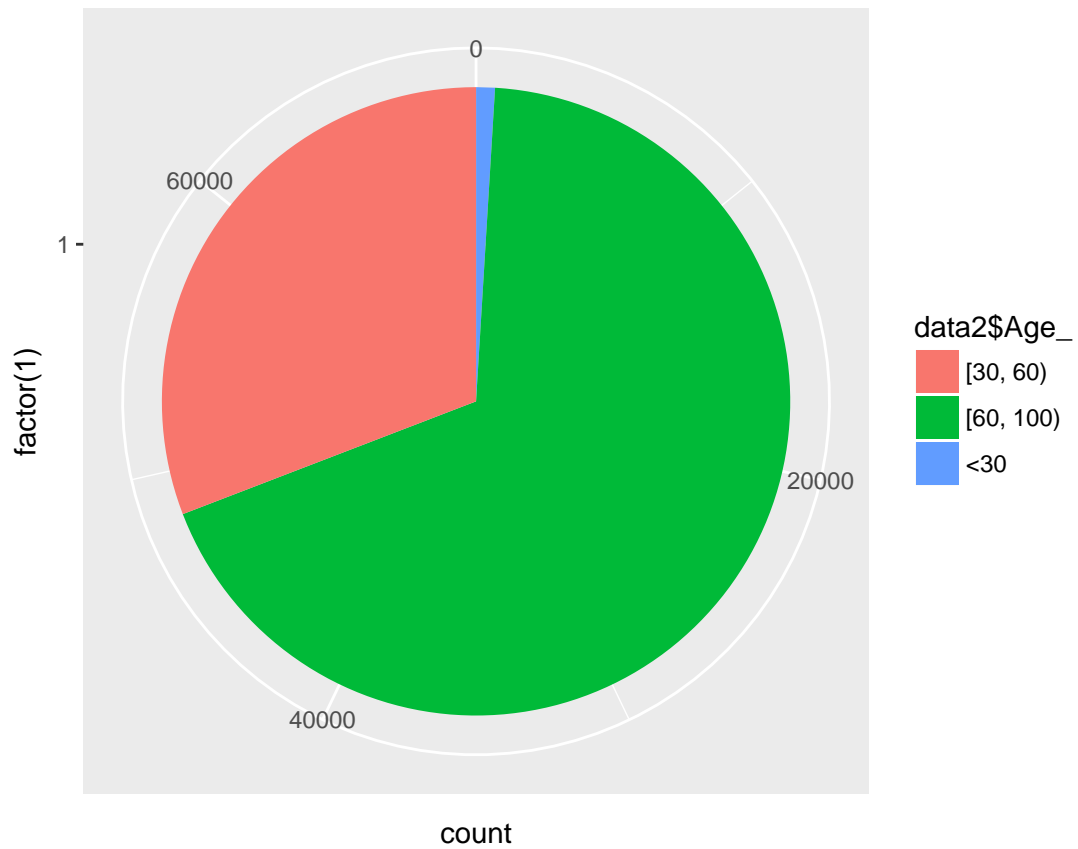
```
# Other          :12821
# Surgery        : 3717
#
#      Pri_diag    Readmit    time_in_hospital
# Circulatory:20894    0:66521    Min.      : 1.000
# Others       :12368    1: 3469    1st Qu.: 2.000
# Respiratory: 9564          Median : 4.000
# Digestive    : 6579          Mean   : 4.302
# Diabetes     : 5660          3rd Qu.: 6.000
# Injury       : 4969          Max.    :14.000
# (Other)      : 9956
```

Data Analysis

```
ggplot(data)+geom_bar(width=1, aes(x=factor(1),fill=data2$Medical_speciality))+coord_polar(theta="y")
```

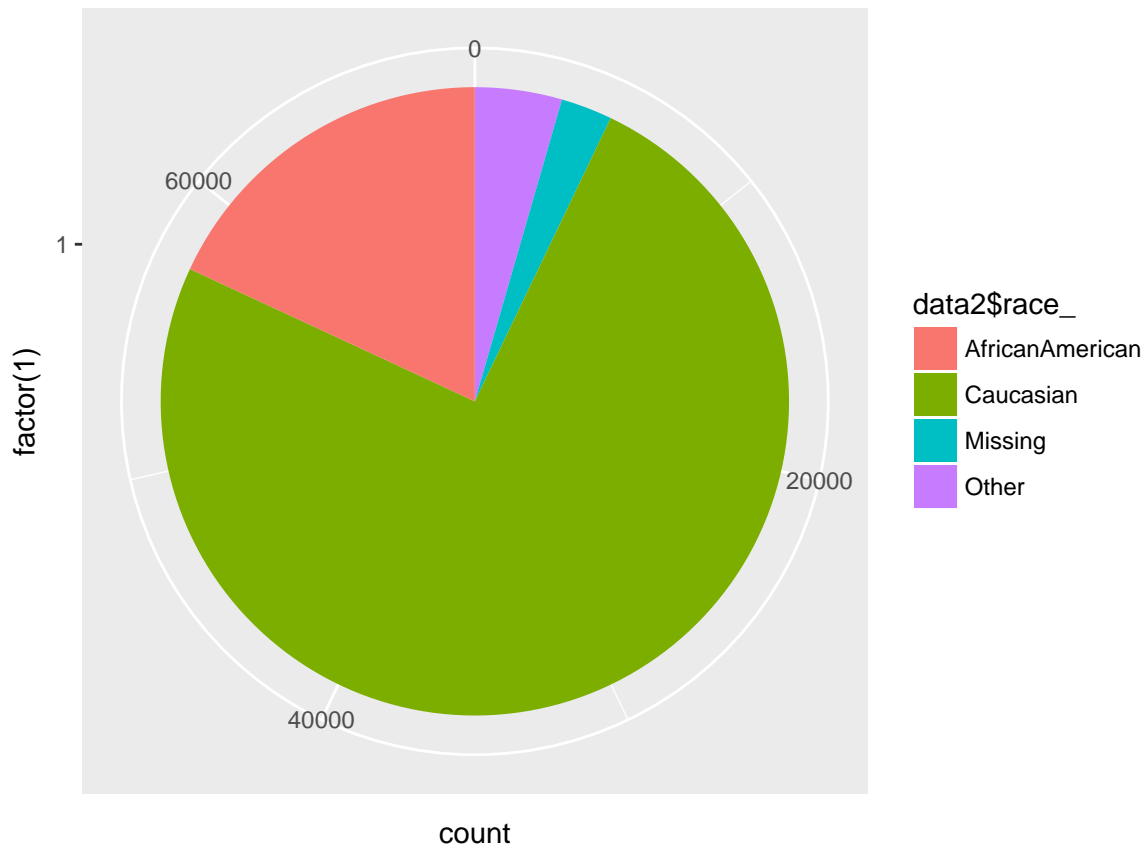


```
ggplot(data)+geom_bar(width=1, aes(x=factor(1),fill=data2$Age_))+coord_polar(theta="y")
```

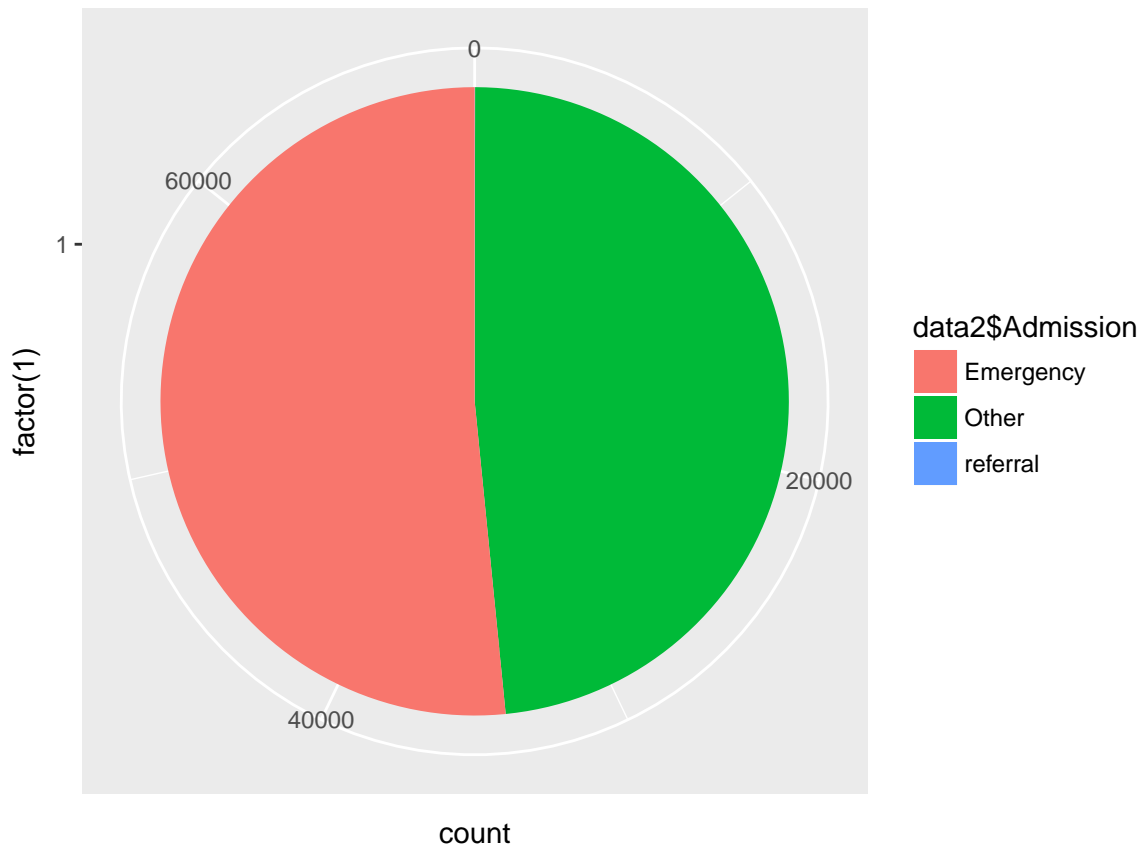


```
ggplot(data)+geom_bar(width=1, aes(x=factor(1),fill=data2$age_))+coord_polar(theta="y")
```

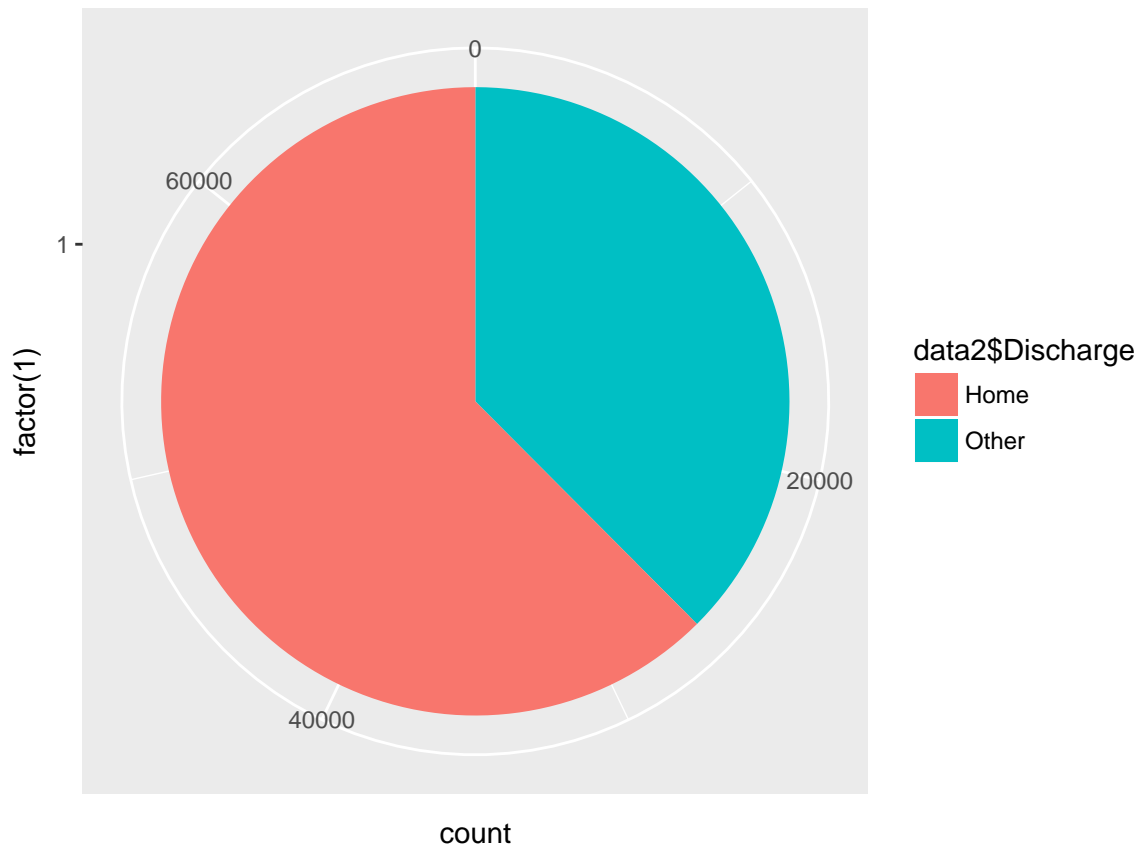




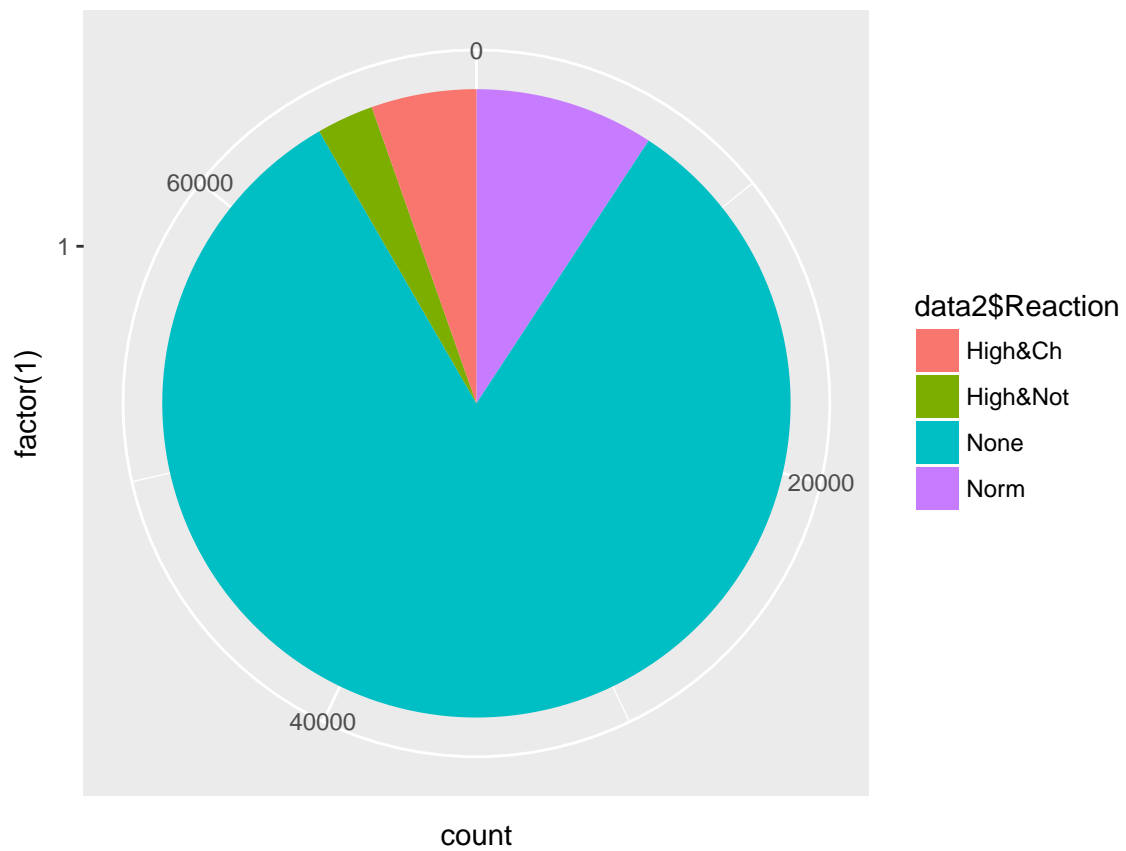
```
ggplot(data)+geom_bar(width=1, aes(x=factor(1),fill=data2$Admission))+coord_polar(theta="y")
```



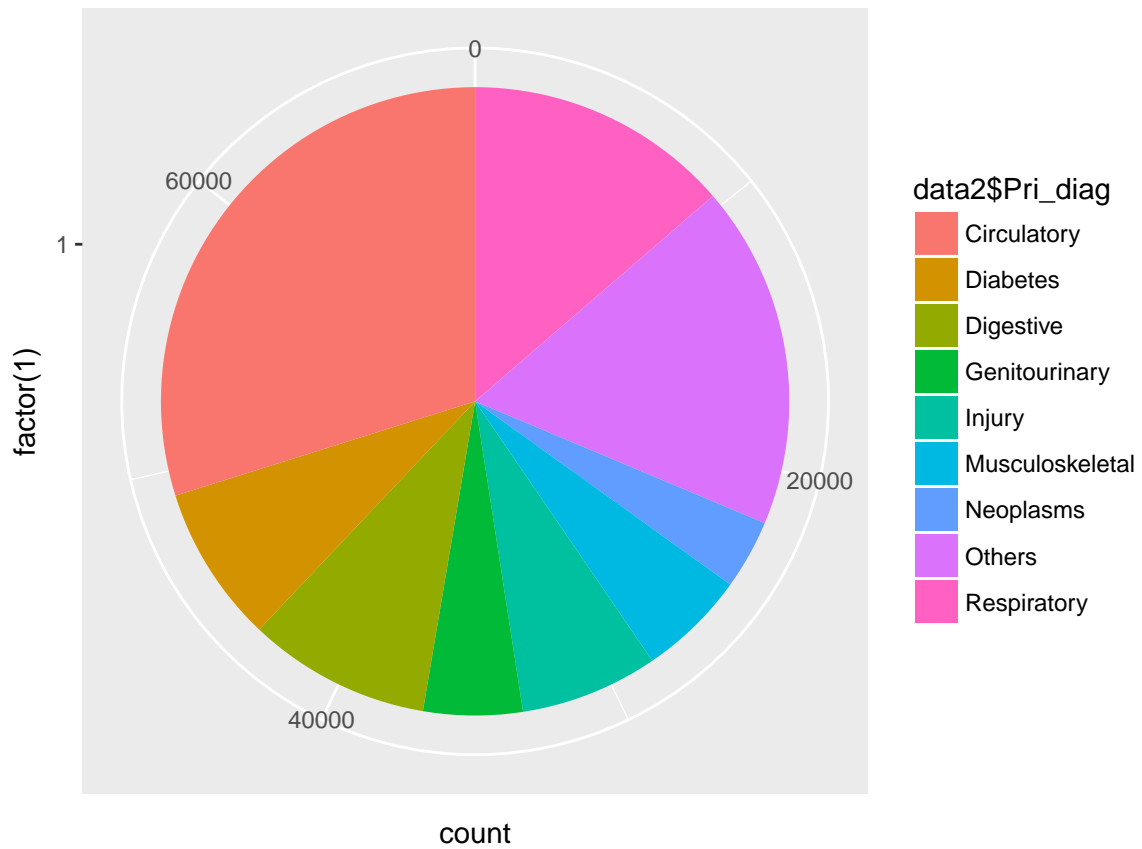
```
ggplot(data)+geom_bar(width=1, aes(x=factor(1),fill=data2$Discharge))+coord_polar(theta="y")
```



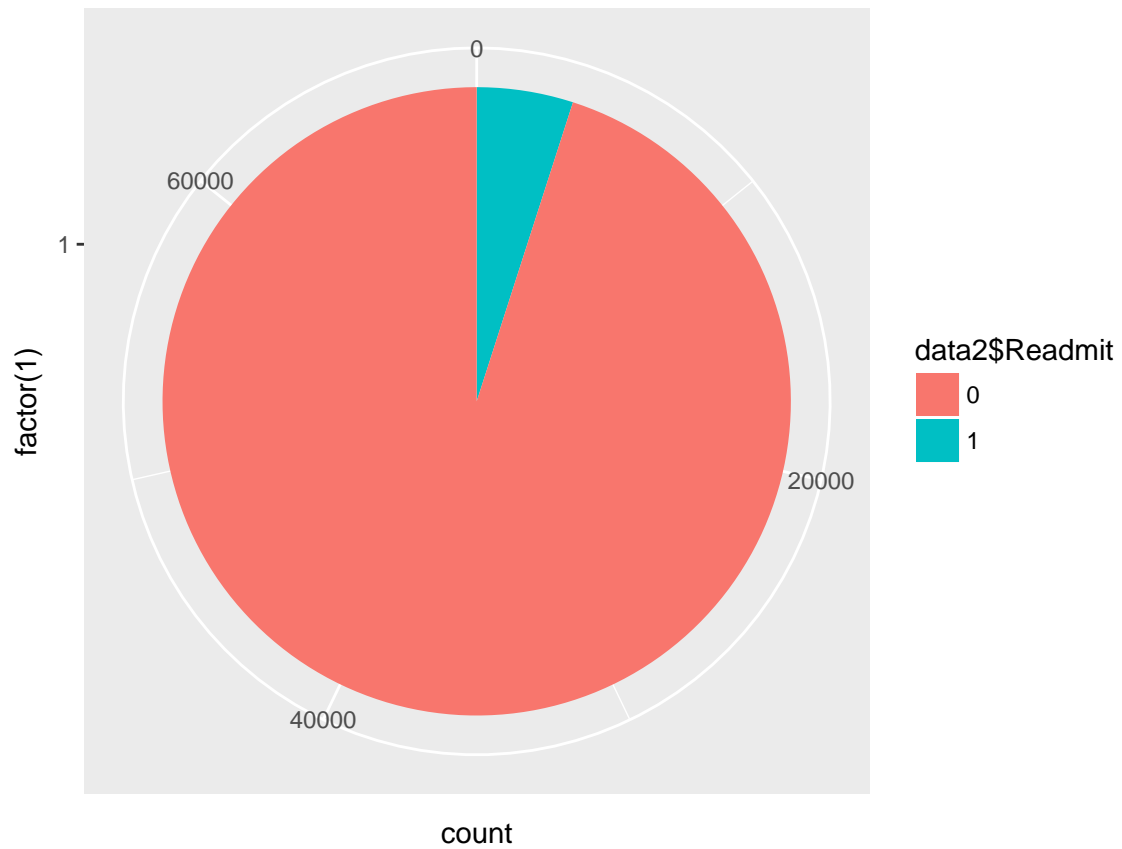
```
ggplot(data)+geom_bar(width=1, aes(x=factor(1),fill=data2$Reaction))+coord_polar(theta="y")
```



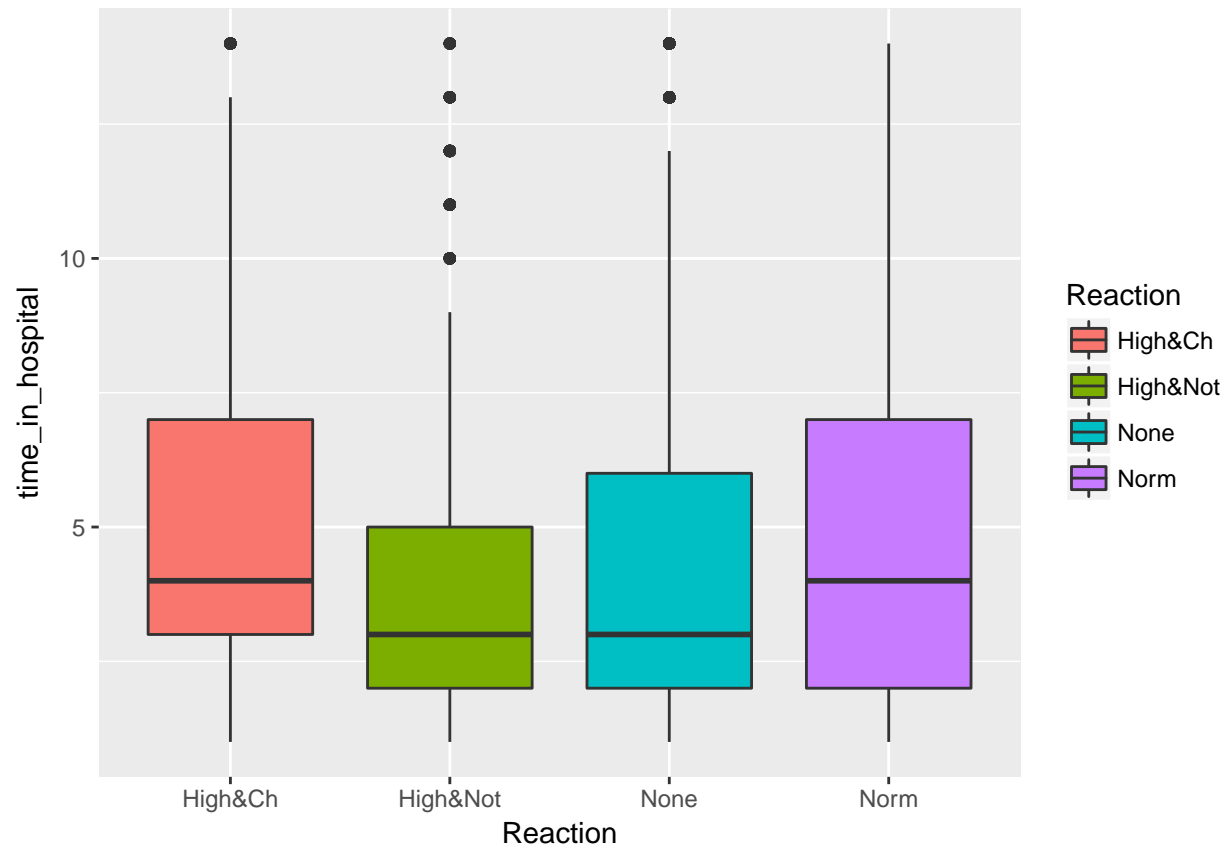
```
ggplot(data)+geom_bar(width=1, aes(x=factor(1),fill=data2$Pri_diag))+coord_polar(theta="y")
```



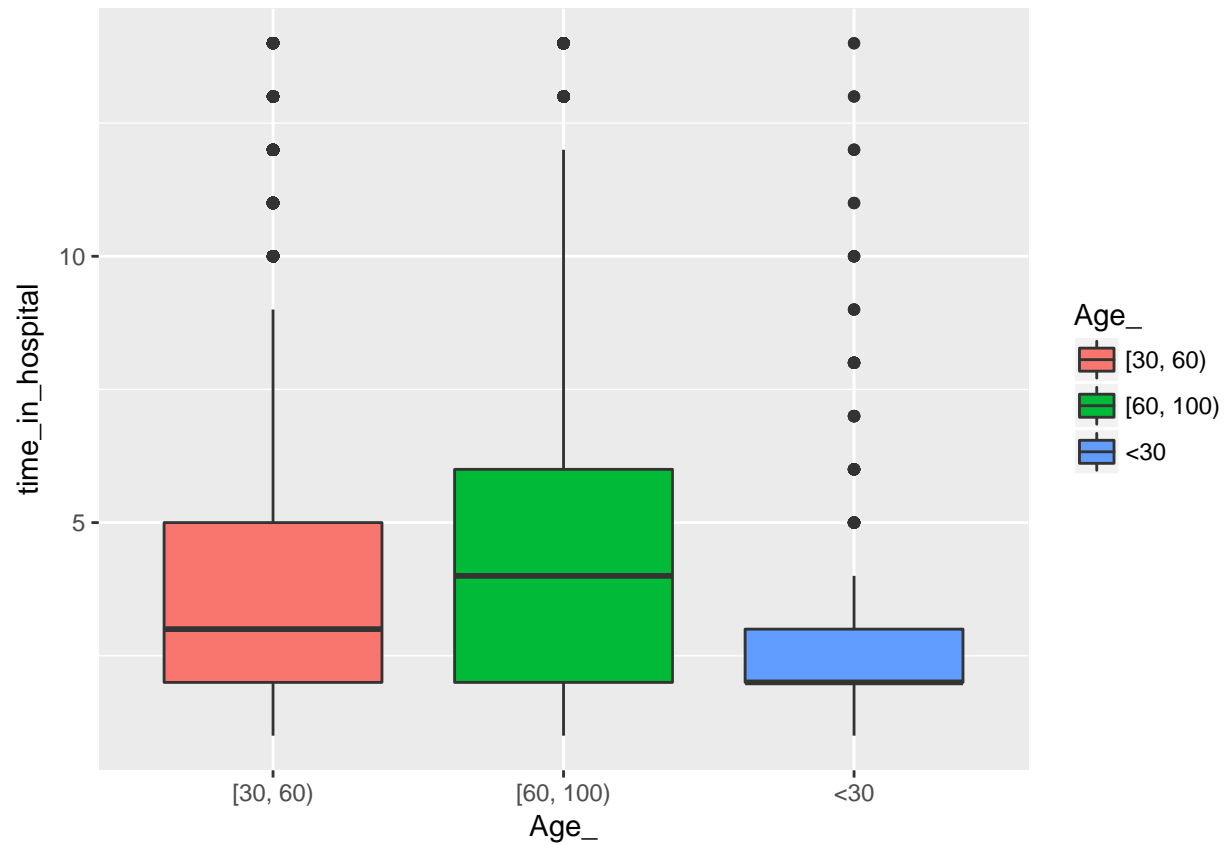
```
ggplot(data)+geom_bar(width=1, aes(x=factor(1),fill=data2$Readmit))+coord_polar(theta="y")
```



```
g <- ggplot(data2,aes(x=Reaction, y=time_in_hospital))  
g + geom_boxplot(aes(fill=Reaction))
```

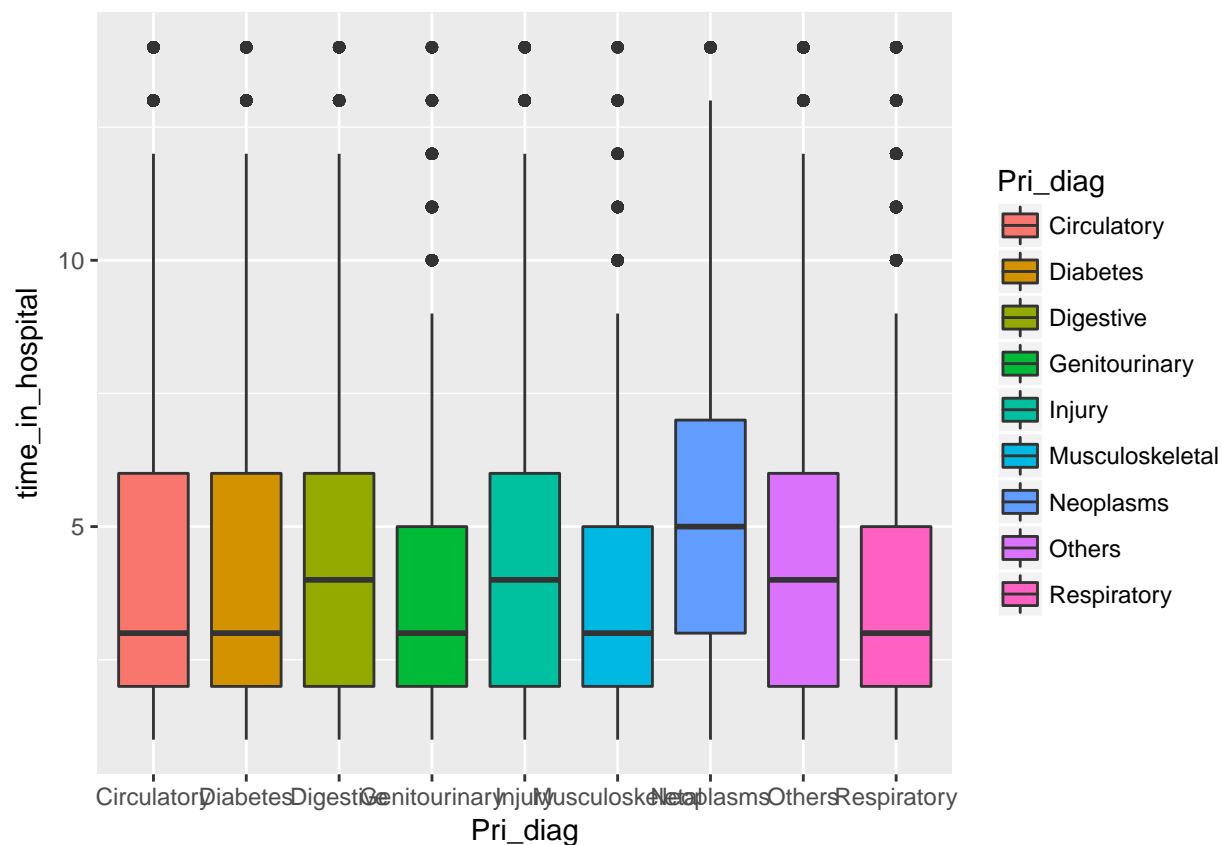


```
g <- ggplot(data2,aes(x=Age_, y=time_in_hospital))  
g + geom_boxplot(aes(fill=Age_))
```



```
g <- ggplot(data2,aes(x=Pri_diag, y=time_in_hospital))
g + geom_boxplot(aes(fill=Pri_diag))
```





Relevel for references

```
data2$race_ <- relevel(data2$race_, ref = 'AfricanAmerican')
data2$Reaction <- relevel(data2$Reaction, ref = 'None')
data2$Pri_diag <- relevel(data2$Pri_diag, ref = 'Diabetes')
```

Basic model, order-one

```
linModel_no2 <- glm(Readmit ~ . , data2, family = binomial)
#summary(linModel_no2)
```

# Coefficients:

#		Estimate	Std. Error	z value	Pr(> z )
# (Intercept)		-3.720905	0.116511	-31.936	< 2e-16 ***
# Medical_specialityGeneralPractice		0.037285	0.102993	0.362	0.717339
# Medical_specialityInternalMedicine		0.299026	0.089461	3.343	0.000830 ***
# Medical_specialityMissing		0.014635	0.083602	0.175	0.861037
# Medical_specialityOther		-0.285336	0.093223	-3.061	0.002207 **
# Medical_specialitySurgery		-0.079796	0.112335	-0.710	0.477490
# Age_[60, 100)		0.348403	0.044247	7.874	3.43e-15 ***
# Age_<30		-0.314197	0.277667	-1.132	0.257819
# race_Caucasian		0.344032	0.052194	6.591	4.36e-11 ***
# race_Missing		-0.114495	0.137427	-0.833	0.404768
# race_Other		0.371437	0.096633	3.844	0.000121 ***
# AdmissionOther		-0.006926	0.038174	-0.181	0.856024
# Admissionreferral		-9.825065	80.300762	-0.122	0.902619
# DischargeOther		0.511007	0.038326	13.333	< 2e-16 ***

```

# ReactionHigh&Ch -0.088169 0.082880 -1.064 0.287413
# ReactionHigh&Not -0.181984 0.118067 -1.541 0.123230
# ReactionNorm -0.127005 0.062829 -2.021 0.043234 *
# Pri_diagCirculatory -0.153162 0.070793 -2.164 0.030500 *
# Pri_diagDigestive -0.272778 0.087827 -3.106 0.001897 **
# Pri_diagGenitourinary -0.297167 0.102871 -2.889 0.003868 **
# Pri_diagInjury -0.031670 0.087207 -0.363 0.716487
# Pri_diagMusculoskeletal -0.238246 0.102064 -2.334 0.019581 *
# Pri_diagNeoplasms -0.104240 0.110581 -0.943 0.345857
# Pri_diagOthers -0.135494 0.074824 -1.811 0.070168 .
# Pri_diagRespiratory -0.268682 0.079986 -3.359 0.000782 ***
# time_in_hospital 0.039181 0.005803 6.752 1.46e-11 ***
# ---
# Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Select all pairwise attributes to train a logit model

```

#linModel_no4 <- glm(Readmit ~ .^2, data2, family = binomial)
#summary(linModel_no4)

```

Reserve the important pairwise attributes and train model We found Age\_\*Medical\_speciality useless, dropped.

```

linModel_core_num <- glm(Readmit ~ Discharge + race_ + Admission + Medical_speciality + time_in_hospital +
  Pri_diag*Discharge + race_*Discharge
  + Discharge*time_in_hospital + Medical_speciality*Discharge
  + time_in_hospital*Medical_speciality
  + time_in_hospital* Pri_diag + Reaction*Pri_diag,
  data = data2, family = binomial)
# summary(linModel_core_num)

# Coefficients:
#
# (Intercept) -4.145212 0.197483 -20.990 < 2e-16 ***
# DischargeOther 0.901349 0.244866 3.681 0.000232 ***
# race_Caucasian 0.437519 0.074212 5.895 3.74e-09 ***
# race_Missing -0.296718 0.210110 -1.412 0.157891
# race_Other 0.302575 0.135545 2.232 0.025596 *
# AdmissionOther -0.005596 0.038397 -0.146 0.884123
# Admissionreferral -9.784584 80.554871 -0.121 0.903323
# Medical_specialityGeneralPractice 0.401829 0.188392 2.133 0.032930 *
# Medical_specialityInternalMedicine 0.500129 0.157556 3.174 0.001502 **
# Medical_specialityMissing 0.363802 0.144066 2.525 0.011562 *
# Medical_specialityOther -0.297803 0.169253 -1.760 0.078490 .
# Medical_specialitySurgery 0.256724 0.213291 1.204 0.228732
# time_in_hospital 0.136690 0.032557 4.198 2.69e-05 ***
# Age_[60, 100) 0.339596 0.044287 7.668 1.75e-14 ***
# Age_<30 -0.157528 0.282762 -0.557 0.577455
# Pri_diagCirculatory -0.037004 0.138808 -0.267 0.789787
# Pri_diagDigestive -0.342643 0.171152 -2.002 0.045287 *
# Pri_diagGenitourinary -0.322777 0.202376 -1.595 0.110725
# Pri_diagInjury -0.009665 0.185387 -0.052 0.958422
# Pri_diagMusculoskeletal -0.703889 0.238734 -2.948 0.003194 **
# Pri_diagNeoplasms 0.221054 0.216383 1.022 0.306976
# Pri_diagOthers -0.027303 0.148768 -0.184 0.854386

```

# Pri_diagRespiratory	-0.425242	0.157461	-2.701	0.006921	**
# ReactionHigh&Ch	-0.750748	0.215940	-3.477	0.000508	***
# ReactionHigh&Not	-0.531529	0.295624	-1.798	0.072178	.
# ReactionNorm	-0.207005	0.211000	-0.981	0.326560	
# DischargeOther:Pri_diagCirculatory	-0.051205	0.145939	-0.351	0.725688	
# DischargeOther:Pri_diagDigestive	0.139439	0.183223	0.761	0.446636	
# DischargeOther:Pri_diagGenitourinary	-0.243134	0.213656	-1.138	0.255133	
# DischargeOther:Pri_diagInjury	0.095687	0.187207	0.511	0.609261	
# DischargeOther:Pri_diagMusculoskeletal	0.453188	0.234151	1.935	0.052935	.
# DischargeOther:Pri_diagNeoplasms	-0.125321	0.228869	-0.548	0.583991	
# DischargeOther:Pri_diagOthers	0.348167	0.154750	2.250	0.024457	*
# DischargeOther:Pri_diagRespiratory	0.253834	0.165773	1.531	0.125716	
# DischargeOther:race_Caucasian	-0.186434	0.103542	-1.801	0.071773	.
# DischargeOther:race_Missing	0.331627	0.278537	1.191	0.233809	
# DischargeOther:race_Other	0.190659	0.193892	0.983	0.325447	
# DischargeOther:time_in_hospital	-0.036059	0.011828	-3.049	0.002299	**
# DischargeOther:Medical_specialityGeneralPractice	-0.121850	0.223700	-0.545	0.585959	
# DischargeOther:Medical_specialityInternalMedicine	-0.121562	0.196055	-0.620	0.535230	
# DischargeOther:Medical_specialityMissing	-0.362101	0.184227	-1.966	0.049355	*
# DischargeOther:Medical_specialityOther	-0.007635	0.205228	-0.037	0.970325	
# DischargeOther:Medical_specialitySurgery	0.503798	0.245500	2.052	0.040157	*
# Medical_specialityGeneralPractice:time_in_hospital	-0.071874	0.033733	-2.131	0.033116	*
# Medical_specialityInternalMedicine:time_in_hospital	-0.037413	0.028476	-1.314	0.188901	
# Medical_specialityMissing:time_in_hospital	-0.041631	0.026728	-1.558	0.119327	
# Medical_specialityOther:time_in_hospital	-0.007006	0.029872	-0.235	0.814562	
# Medical_specialitySurgery:time_in_hospital	-0.133988	0.038119	-3.515	0.000440	***
# time_in_hospital:Pri_diagCirculatory	-0.041732	0.021838	-1.911	0.056013	.
# time_in_hospital:Pri_diagDigestive	-0.017480	0.027729	-0.630	0.528445	
# time_in_hospital:Pri_diagGenitourinary	0.011861	0.033335	0.356	0.721985	
# time_in_hospital:Pri_diagInjury	-0.030957	0.027655	-1.119	0.262977	
# time_in_hospital:Pri_diagMusculoskeletal	0.009490	0.036182	0.262	0.793098	
# time_in_hospital:Pri_diagNeoplasms	-0.072089	0.033778	-2.134	0.032824	*
# time_in_hospital:Pri_diagOthers	-0.080533	0.022926	-3.513	0.000443	***
# time_in_hospital:Pri_diagRespiratory	-0.009011	0.024637	-0.366	0.714570	
# Pri_diagCirculatory:ReactionHigh&Ch	1.044042	0.250548	4.167	3.09e-05	***
# Pri_diagDigestive:ReactionHigh&Ch	0.286802	0.507700	0.565	0.572139	
# Pri_diagGenitourinary:ReactionHigh&Ch	0.819908	0.433084	1.893	0.058333	.
# Pri_diagInjury:ReactionHigh&Ch	0.190152	0.509654	0.373	0.709075	
# Pri_diagMusculoskeletal:ReactionHigh&Ch	0.880868	0.566935	1.554	0.120247	
# Pri_diagNeoplasms:ReactionHigh&Ch	0.249116	0.761235	0.327	0.743477	
# Pri_diagOthers:ReactionHigh&Ch	0.636597	0.299036	2.129	0.033268	*
# Pri_diagRespiratory:ReactionHigh&Ch	0.791565	0.316789	2.499	0.012465	*
# Pri_diagCirculatory:ReactionHigh&Not	0.441651	0.351824	1.255	0.209364	
# Pri_diagDigestive:ReactionHigh&Not	0.135432	0.591980	0.229	0.819042	
# Pri_diagGenitourinary:ReactionHigh&Not	0.383801	0.666438	0.576	0.564684	
# Pri_diagInjury:ReactionHigh&Not	0.671458	0.555651	1.208	0.226888	
# Pri_diagMusculoskeletal:ReactionHigh&Not	1.039674	0.610377	1.703	0.088506	.
# Pri_diagNeoplasms:ReactionHigh&Not	1.523366	0.694452	2.194	0.028263	*
# Pri_diagOthers:ReactionHigh&Not	0.010094	0.467974	0.022	0.982792	
# Pri_diagRespiratory:ReactionHigh&Not	0.387420	0.454992	0.851	0.394498	
# Pri_diagCirculatory:ReactionNorm	0.040582	0.238436	0.170	0.864851	
# Pri_diagDigestive:ReactionNorm	0.152586	0.322777	0.473	0.636408	
# Pri_diagGenitourinary:ReactionNorm	0.104212	0.366184	0.285	0.775959	

```

# Pri_diagInjury:ReactionNorm          0.015613    0.319722    0.049 0.961051
# Pri_diagMusculoskeletal:ReactionNorm  0.133092    0.375174    0.355 0.722779
# Pri_diagNeoplasms:ReactionNorm        0.347555    0.419031    0.829 0.406864
# Pri_diagOthers:ReactionNorm           0.266480    0.249662    1.067 0.285807
# Pri_diagRespiratory:ReactionNorm      -0.161596    0.278172   -0.581 0.561294
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Chisq test

```

# anova(linModel_core_num, test="Chisq")
#
#
#              Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
# NULL                                69989      27608
# Discharge              1    365.64    69988      27243 < 2.2e-16 ***
# race_                   3     56.68    69985      27186 3.001e-12 ***
# Admission               2      6.11    69983      27180 0.047043 *
# Medical_speciality      5     99.47    69978      27080 < 2.2e-16 ***
# time_in_hospital       1     50.42    69977      27030 1.243e-12 ***
# Age_                   2     69.32    69975      26961 8.844e-16 ***
# Pri_diag                8     22.74    69967      26938 0.003709 **
# Reaction               3      7.00    69964      26931 0.071785 .
# Discharge:Pri_diag      8     25.27    69956      26906 0.001400 **
# Discharge:race_         3     11.16    69953      26894 0.010881 *
# Discharge:time_in_hospital 1      9.36    69952      26885 0.002214 **
# Discharge:Medical_speciality 5     33.35    69947      26852 3.214e-06 ***
# Medical_speciality:time_in_hospital 5     19.96    69942      26832 0.001271 **
# time_in_hospital:Pri_diag 8     25.81    69934      26806 0.001132 **
# Pri_diag:Reaction      24     33.91    69910      26772 0.086257 .
#
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Graph-probability of readmission based on reference value create reference data frame & calculate fit/confidence interval

```

time_hos = 4.3
Diag_list = list("Diabetes","Respiratory","Circulatory","Digestive","Others")
Diag_list2 = list()
Readmit_list = list("None","Norm","High&Ch","High&Not")

TEST = list(list(),list(),list(),list(),list())
TEST2 = list(list(),list(),list(),list(),list())
for (i in 1:5) {
  for (j in 1:4){
    temp = data.frame( "Medical_speciality" = "Cardiology" , "Age_" = as.factor("[30, 60)"), "race_" =
    TEST[[i]][[j]] = temp
    colnames(TEST[[i]][[j]]) <- c("Medical_speciality","Age_","race_","Admission","Discharge","Reaction")
  }
}

critval <- 1.96 ## approx 95% CI
plot_data = list(c(),c(),c(),c(),c())
upr_data = list(c(),c(),c(),c(),c())
lwr_data = list(c(),c(),c(),c(),c())
for (i in 1:5 ){

```

```

for (j in 1:4){
  temp = predict(linModel_core_num, TEST[[i]][[j]], type="response", se.fit = TRUE)
  plot_data[[i]][j] = temp$fit
  upr_data[[i]][j] = temp$fit + (critval * temp$se.fit)
  lwr_data[[i]][j] = temp$fit - (critval * temp$se.fit)
}
}

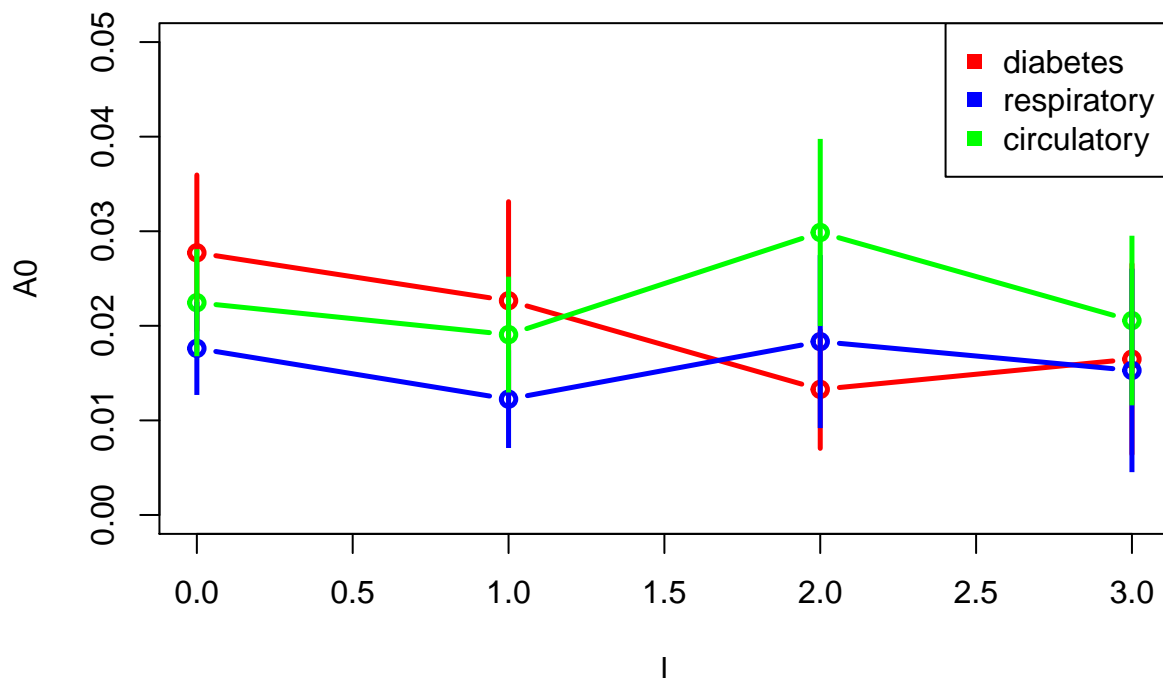
```

plot graph

```

A0 = plot_data[[1]]
B0 = plot_data[[2]]
C0 = plot_data[[3]]
D0 = plot_data[[4]]
E0 = plot_data[[5]]
l = c(0,1,2,3)
plot(l,A0, type="b", ylim = c(0, 0.05),col="red",lwd=2.5)
lines(l,B0,type = "b",col="blue",lwd=2.5)
lines(l,C0,type = "b",col="green",lwd=2.5)
legend("topright", c("diabetes", "respiratory", "circulatory"), col = c("red", "blue", "green"), pch = c(1,2,3))
segments(x0=l, y0=lwr_data[[1]], y1= upr_data[[1]], col="red",lwd=2.5,lend=0)
segments(x0=l, y0=lwr_data[[2]], y1=upr_data[[2]],col="blue",lwd=2.5,lend=1)
segments(x0=l, y0=lwr_data[[3]], y1=upr_data[[3]],col="green",lwd=2.5,lend=2)

```



```

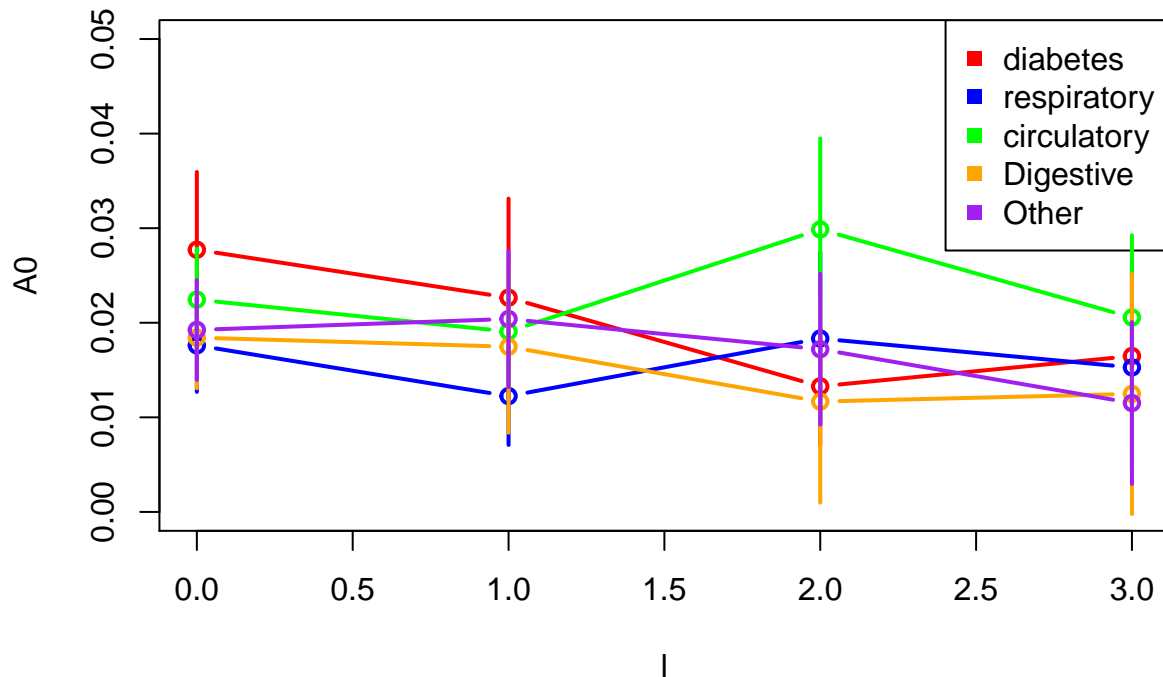
plot( l,A0, type="b", ylim = c(0, 0.05),col="red",lwd=2)
lines(l,B0,type = "b",col="blue",lwd=2)
lines(l,C0,type = "b",col="green",lwd=2)

```

```

lines(1,D0,type = "b",col="orange",lwd=2)
lines(1,E0,type = "b",col="purple",lwd=2)
legend("topright", c("diabetes", "respiratory", "circulatory", "Digestive", "Other"), col = c("red", "blue", "green", "orange", "purple"),
segments(x0=1, y0=lwr_data[[1]], y1=upr_data[[1]],col="red",lwd=2)
segments(x0=1, y0=lwr_data[[2]], y1=upr_data[[2]],col="blue",lwd=2)
segments(x0=1, y0=lwr_data[[3]], y1=upr_data[[3]],col="green",lwd=2)
segments(x0=1, y0=lwr_data[[4]], y1=upr_data[[4]],col="orange",lwd=2)
segments(x0=1, y0=lwr_data[[5]], y1=upr_data[[5]],col="purple",lwd=2)

```



age: 30-60 -> 60-100 Home -> Other change in probability

```

time_hos = 4.3
Diag_list = list("Diabetes","Respiratory","Circulatory","Digestive","Others")
Readmit_list = list("None","Norm","High&Ch","High&Not")

TEST2 = list(list(),list(),list(),list(),list())
for (i in 1:5) {
  for (j in 1:4){
    temp = data.frame( "Medical_speciality" = "Cardiology" , "Age_" = as.factor("[60, 100)"), "race_" = 
    TEST2[[i]][[j]] = temp
    colnames(TEST2[[i]][[j]]) <- c("Medical_speciality","Age_","race_","Admission","Discharge","Reaction")
  }
}

critval <- 1.96 ## approx 95% CI
plot_data2 = list(c(),c(),c(),c(),c())

```

```

upr_data2 = list(c(),c(),c(),c(),c())
lwr_data2 = list(c(),c(),c(),c(),c())
for (i in 1:5){
  for (j in 1:4){
    temp = predict(linModel_core_num, TEST2[[i]][[j]], type="response",se.fit = TRUE)
    plot_data2[[i]][j] = temp$fit
    upr_data2[[i]][j] = temp$fit + (critval * temp$se.fit)
    lwr_data2[[i]][j] = temp$fit - (critval * temp$se.fit)
  }
}

```

```

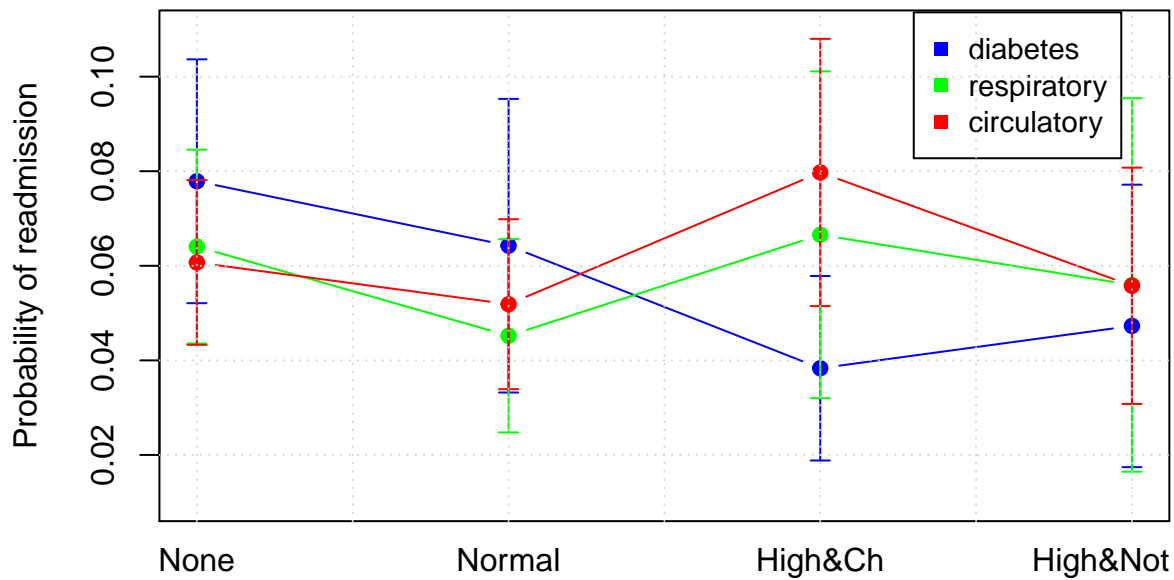
A02 = plot_data2[[1]]
B02 = plot_data2[[2]]
C02 = plot_data2[[3]]
D02 = plot_data2[[4]]
E02 = plot_data2[[5]]
lr = 1

```

```

plot( 1,A02,ann=F , type="b", ylim = c(0.01, 0.11),col="blue",lwd=lr,pch=20,cex=1.5, xaxt="n")
title(ylab = 'Probability of readmission')
mtext("None",side=1,at=0,line=0.5)
mtext("Normal",side=1,at=1,line=0.5)
mtext("High&Ch",side=1,at=2,line=0.5)
mtext("High&Not",side=1,at=2.9,line=0.5)
lines(1,B02,type = "b",col="green",lwd=lr,pch=20,cex=1.5)
lines(1,C02,type = "b",col="red",lwd=lr,pch=20,cex=1.5)
legend(2.3,0.1136, c("diabetes", "respiratory", "circulatory"), col = c("blue", "green", "red"), pch = 20)
arrows(1,lwr_data2[[1]],1,upr_data2[[1]],code=3,col="blue",lwd=lr,length= 0.05,angle=90)
arrows(1,lwr_data2[[2]],1,upr_data2[[2]],code=3,col="green",lwd=lr,length= 0.05,angle=90)
arrows(1,lwr_data2[[3]],1,upr_data2[[3]],code=3,col="red",lwd=lr,length= 0.05,angle=90)
grid()

```



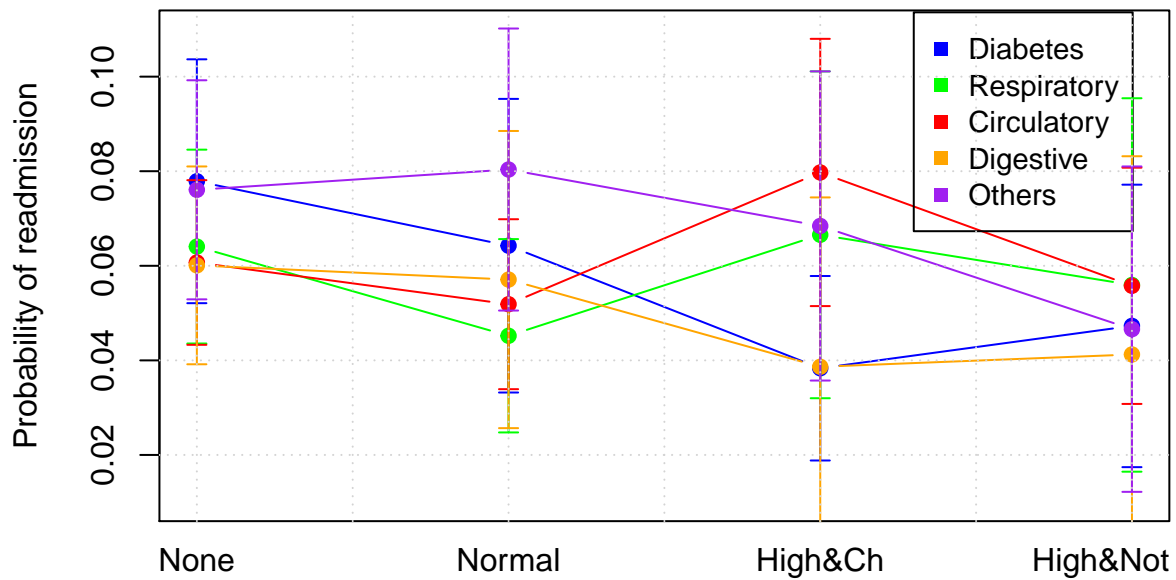
```
A02 = plot_data2[[1]]
B02 = plot_data2[[2]]
C02 = plot_data2[[3]]
D02 = plot_data2[[4]]
E02 = plot_data2[[5]]
lr = 1

plot( 1,A02,ann=F , type="b", ylim = c(0.01, 0.11),col="blue",lwd=lr,pch=20,cex=1.5, xaxt="n")
title(ylab = 'Probability of readmission')
mtext("None",side=1,at=0,line=0.5)
mtext("Normal",side=1,at=1,line=0.5)
mtext("High&Ch",side=1,at=2,line=0.5)
mtext("High&Not",side=1,at=2.9,line=0.5)
lines(1,B02,type = "b",col="green",lwd=lr,pch=20,cex=1.5)
lines(1,C02,type = "b",col="red",lwd=lr,pch=20,cex=1.5)
lines(1,D02,type = "b",col="orange",lwd=lr,pch=20,cex=1.5)
lines(1,E02,type = "b",col="purple",lwd=lr,pch=20,cex=1.5)

legend(2.3,0.1136, c("Diabetes","Respiratory","Circulatory","Digestive","Others"), col = c("blue", "green", "red", "orange", "purple"),
arrows(1,lwr_data2[[1]],1,upr_data2[[1]],code=3,col="blue",lwd=lr,length= 0.05,angle=90)
arrows(1,lwr_data2[[2]],1,upr_data2[[2]],code=3,col="green",lwd=lr,length= 0.05,angle=90)
arrows(1,lwr_data2[[3]],1,upr_data2[[3]],code=3,col="red",lwd=lr,length= 0.05,angle=90)
arrows(1,lwr_data2[[4]],1,upr_data2[[4]],code=3,col="orange",lwd=lr,length= 0.05,angle=90)
arrows(1,lwr_data2[[5]],1,upr_data2[[5]],code=3,col="purple",lwd=lr,length= 0.05,angle=90)

grid()
```





#### Test and Other Methods

Comment: This data set is a super-unbalanced one, therefore accuracy of all these methods are meaningless. However, good to learn!

Split data\_full into train and test set

```
data_full$Diag1 = as.factor(data_full$Diag1)
data_full$Diag2 = as.factor(data_full$Diag2)
data_full$Diag3 = as.factor(data_full$Diag3)
data_full$HbA1c = as.factor(data_full$HbA1c)

data_full = data2
data_full$Readmit = as.factor(data_full$Readmit)
colnames(data_full)

## [1] "Medical_speciality" "Age_" "race_"
## [4] "Admission" "Discharge" "Reaction"
## [7] "Pri_diag" "Readmit" "time_in_hospital"

set.seed(17)
inTrain <- createDataPartition(y = data_full$Readmit, p = .60, list = FALSE)
train <- data_full[inTrain,]
test <- data_full[-inTrain,]
nrow(train)

## [1] 41995
```

```
nrow(test)
```

```
## [1] 27995
```

Test of logistic model

```
test$pred_readmit <- predict(linModel_core_num, test, type="response")
# test$pred_readmit = as.numeric(test$pred_readmit)
# test$pred_readmit[test$pred_readmit>=0.5] <- 1
# test$pred_readmit[test$pred_readmit<0.5] <- 0
# test$pred_readmit = as.factor(as.integer(test$pred_readmit))
summary(test$pred_readmit)
```

```
##      Min.   1st Qu.   Median     Mean   3rd Qu.    Max.
## 3.080e-06 3.093e-02 4.696e-02 4.949e-02 6.439e-02 1.922e-01
```

```
# confusionMatrix(test$pred_readmit, test$Readmit)
```

```
#Accuracy : 0.9505
```

```
#      Min.   1st Qu.   Median     Mean   3rd Qu.    Max.
#1.600e-07 3.117e-02 4.695e-02 4.949e-02 6.434e-02 1.900e-01
```

Random Forest

```
#
#Rf_fit<-randomForest(formula=Readmit ~ Medical_speciality+Age_+race_+Admission+Discharge+Reaction+Pri_
# Rf_fit<-randomForest(formula=Readmit ~ .,
#                       data=train, ntree=500)
#
# print(Rf_fit)
#
# test$pred_readmit <- predict(Rf_fit, test, type = "response")
# table(test$Readmit, test$pred_readmit)
# prop.table(table(test$Readmit, test$pred_readmit),1)

# #importance(Rf_fit)
```

Naive Bayes

```
# # build decision tree with naive Bayes in the leaves
# nbModel <- CoreModel("Readmit", train, model="tree", modelType=4)
# #print(nbModel)
#
# # prediction on test set
# pred <- predict(nbModel, test, type="class")
# # mEval <- modelEval(nbModel, test$Readmit, pred)
# # print(mEval) # evaluation of the model
#
# test$pred_readmit <- pred
# prop.table(table(test$Readmit, test$pred_readmit),1)
# confusionMatrix(test$pred_readmit, test$Readmit)

#Accuracy : 0.949
```

SVM

```

# SVMmodel <- svm(Readmit~ .,
#               data=train, kernel = "linear")
# print(SVMmodel)
# summary(SVMmodel)
# x <- select(test, -Readmit)
# y <- select(test, Readmit)
# pred <- predict(SVMmodel, x)
# test$pred_readmit <- pred
# prop.table(table(test$Readmit, test$pred_readmit),1)
# confusionMatrix(test$pred_readmit, test$Readmit)

# Accuracy : 0.9505

```

## Neural Networks

```

# nnet_model <- nnet(formula = Readmit ~ ., data=train, size = 10, maxit = 100)
#
# test$pred_readmit <- predict(nnet_model, test, type = "class")
# test$pred_readmit = as.factor(test$pred_readmit)
# prop.table(table(test$Readmit, test$pred_readmit),1)
#
# #summary(nnet_model)
#
# confusionMatrix(test$pred_readmit, test$Readmit)
# Accuracy : 0.9499

```

## Rpart Tree

```

# rpart_tree <- rpart(formula = Readmit ~ .,
#                   data=train, method = 'class')
# summary(rpart_tree)
#
# test$pred_readmit <- predict(rpart_tree, test, type="class")
# table(predict(rpart_tree, test, type="class"), test$Readmit)
# prop.table(table(test$Readmit, test$pred_readmit),1)
#
# confusionMatrix(test$pred_readmit, test$Readmit)

# Accuracy : 0.9505

```

## KNN

```

# KNN
# knnModel <- CoreModel(Readmit ~ .,
#                   data=train, model="knn", kInNN = 5)
# print(knnModel)
#
# pred <- predict(knnModel, test, type="class")
# mEval <- modelEval(knnModel, test$Readmit, pred)
# print(mEval) # evaluation of the model
#
# test$pred_readmit <- pred
# prop.table(table(test$Readmit, test$pred_readmit),1)
# confusionMatrix(test$pred_readmit, test$Readmit)

#Accuracy : 0.9474

```

---

Reference:

[1] Beata Strack,1 Jonathan P. DeShazo Impact of HbA1c Measurement on Hospital Readmission Rates: Analysis of 70,000 Clinical Database Patient Records, 2014. Print. <https://www.hindawi.com/journals/bmri/2014/781670/> [2] Diabetes 130-US hospitals for years 1999-2008 Data Set <https://archive.ics.uci.edu/ml/datasets/Diabetes+130-US+hospitals+for+years+1999-2008#> [3] [https://github.com/tobiolatunji/Readmission\\_Prediction](https://github.com/tobiolatunji/Readmission_Prediction) [4] <https://github.com/swengzju/Predicting-Diabetes-Patient-Readmission> [5] <http://r-statistics.co/Top50-Ggplot2-Visualizations-MasterList-R-Code.html> [6] <https://www.r-bloggers.com/evaluating-logistic-regression-models/>