OZONE

Hafid Pradipta

December 4, 2017

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```r
#ozone dataset
library(readr)
onehr <- read_csv("C:/Users/Hafid Pradipta/OneDrive/STAT 520/dataset/Ozone/onehr.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_character(),
##   t_pk = col_double(),
##   t_av = col_double(),
##   percp = col_double(),
##   attribute = col_integer()
## )

## See spec(...) for full column specifications.

## Warning in rbind(names(probs), probs_f): number of columns of result is not
## a multiple of vector length (arg 1)

## Warning: 352 parsing failures.
## row # A tibble: 5 x 5 col     row   col expected actual expected   <int> <chr>    <chr>  <chr> actual
## ... ................. ... ........................... ........ ........................... .....
## See problems(...) for more details.
```

```r
head(onehr)
```

```
## # A tibble: 6 x 74
##       Date  wsr0  wsr1  wsr2  wsr3  wsr4  wsr5  wsr6  wsr7  wsr8  wsr9
##      <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr>
## 1 1/1/1998   0.8   1.8   2.4   2.1     2   2.1   1.5   1.7   1.9   2.3
## 2 1/2/1998   2.8   3.2   3.3   2.7   3.3   3.2   2.9   2.8   3.1   3.4
## 3 1/3/1998   2.9   2.8   2.6   2.1   2.2   2.5   2.5   2.7   2.2   2.5
## 4 1/4/1998   4.7   3.8   3.7   3.8   2.9   3.1   2.8   2.5   2.4   3.1
## 5 1/5/1998   2.6   2.1   1.6   1.4   0.9   1.5   1.2   1.4   1.3   1.4
## 6 1/6/1998   3.1   3.5   3.3   2.5   1.6   1.7   1.6   1.6   2.3   1.8
## # ... with 63 more variables: wsr10 <chr>, wsr11 <chr>, wsr12 <chr>,
## #   wsr13 <chr>, wsr14 <chr>, wsr15 <chr>, wsr16 <chr>, wsr17 <chr>,
## #   wsr18 <chr>, wsr19 <chr>, wsr20 <chr>, wsr21 <chr>, wsr22 <chr>,
## #   wsr23 <chr>, wsr_pk <chr>, wsr_av <chr>, t0 <chr>, t1 <chr>, t2 <chr>,
## #   t3 <chr>, t4 <chr>, t5 <chr>, t6 <chr>, t7 <chr>, t8 <chr>, t9 <chr>,
## #   t10 <chr>, t11 <chr>, t12 <chr>, t13 <chr>, t14 <chr>, t15 <chr>,
## #   t16 <chr>, t17 <chr>, t18 <chr>, t19 <chr>, t20 <chr>, t21 <chr>,
## #   t22 <chr>, t23 <chr>, t_pk <dbl>, t_av <dbl>, t85 <chr>, rh85 <chr>,
```

```
## #   u85 <chr>, v85 <chr>, ht85 <chr>, t70 <chr>, rh70 <chr>, u70 <chr>,
## #   v70 <chr>, ht70 <chr>, t50 <chr>, rh50 <chr>, u50 <chr>, v50 <chr>,
## #   ht50 <chr>, ki <chr>, tt <chr>, slp <chr>, slp_ <chr>, percp <dbl>,
## #   attribute <int>
```

```
### cleaning the data
which(onehr$wsr0=="?")
```

```
##   [1]  270  277  278  279  280  281  282  283  284  285  286  287  288  289
##  [15]  290  291  292  293  294  295  296  297  298  299  303  559  627 1307
##  [29] 1310 1312 1320 1327 1330 1339 1340 1601 1618 1632 1633 1634 1635 1636
##  [43] 1637 1638 1639 1640 1641 1642 1643 1644 1645 1646 1647 1648 1649 1650
##  [57] 1651 1652 1653 1654 1655 1656 1657 1658 1659 1660 1661 1662 1663 1664
##  [71] 1665 1666 1667 1668 1669 1670 1671 1672 1673 1674 1675 1676 1677 1678
##  [85] 1679 1680 1681 1682 1683 1684 1685 1686 1687 1688 1689 1690 1691 1692
##  [99] 1693 1694 1695 1696 1697 1698 1699 1700 1701 1702 1703 1704 1705 1706
## [113] 1707 1708 1709 1710 1711 1712 1713 1714 1715 1716 1717 1718 1719 1720
## [127] 1721 1722 1723 1724 1725 1726 1727 1728 1729 1730 1731 1732 1733 1734
## [141] 1735 1736 1737 1738 1739 1740 1741 1742 1743 1744 1745 1746 1747 1748
## [155] 1749 1750 1751 1752 1753 1754 1755 1756 1757 1758 1759 1760 1761 1762
## [169] 1763 1764 1765 1766 1767 1768 1769 1770 1771 1772 1773 1774 1775 1776
## [183] 1777 1778 1779 1780 1781 1782 1783 1784 1785 1786 1787 1788 1789 1790
## [197] 1791 1792 1793 1794 1795 1796 1797 1798 1799 1800 1801 1802 1803 1804
## [211] 1805 1806 1807 1808 1809 1810 1811 1812 1813 1814 1815 1816 1817 1818
## [225] 1819 1820 1821 1822 1823 1824 1825 1826 1836 1837 1838 1839 1840 1841
## [239] 1842 1843 1844 1845 1846 1847 1848 1853 1854 1855 1856 1857 1858 1859
## [253] 1860 1861 1862 1863 1864 1865 1866 1867 1868 1869 1870 1871 1872 1873
## [267] 1874 1875 1876 1877 2017 2128 2130 2131 2132 2133 2134 2135 2136 2137
## [281] 2138 2139 2140 2141 2142 2143 2144 2145 2146 2147 2148 2149 2150 2151
## [295] 2152 2153 2154 2155 2156
```

```
onehr[onehr=="?"]<-NA
head(onehr==NA)
```

```
##      Date wsr0 wsr1 wsr2 wsr3 wsr4 wsr5 wsr6 wsr7 wsr8 wsr9 wsr10 wsr11
## [1,]   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA    NA    NA
## [2,]   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA    NA    NA
## [3,]   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA    NA    NA
## [4,]   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA    NA    NA
## [5,]   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA    NA    NA
## [6,]   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA    NA    NA
##      wsr12 wsr13 wsr14 wsr15 wsr16 wsr17 wsr18 wsr19 wsr20 wsr21 wsr22
## [1,]    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA
## [2,]    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA
## [3,]    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA
## [4,]    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA
## [5,]    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA
## [6,]    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA
##      wsr23 wsr_pk wsr_av t0 t1 t2 t3 t4 t5 t6 t7 t8 t9 t10 t11 t12 t13 t14
## [1,]    NA     NA     NA NA NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA
## [2,]    NA     NA     NA NA NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA
## [3,]    NA     NA     NA NA NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA
## [4,]    NA     NA     NA NA NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA
## [5,]    NA     NA     NA NA NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA
## [6,]    NA     NA     NA NA NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA
##      t15 t16 t17 t18 t19 t20 t21 t22 t23 t_pk t_av t85 rh85 u85 v85 ht85
```

```
## [1,]  NA  NA  NA  NA  NA  NA  NA  NA  NA   NA   NA  NA   NA  NA  NA   NA
## [2,]  NA  NA  NA  NA  NA  NA  NA  NA  NA   NA   NA  NA   NA  NA  NA   NA
## [3,]  NA  NA  NA  NA  NA  NA  NA  NA  NA   NA   NA  NA   NA  NA  NA   NA
## [4,]  NA  NA  NA  NA  NA  NA  NA  NA  NA   NA   NA  NA   NA  NA  NA   NA
## [5,]  NA  NA  NA  NA  NA  NA  NA  NA  NA   NA   NA  NA   NA  NA  NA   NA
## [6,]  NA  NA  NA  NA  NA  NA  NA  NA  NA   NA   NA  NA   NA  NA  NA   NA
##      t70 rh70 u70 v70 ht70 t50 rh50 u50 v50 ht50 ki tt slp slp_ percp
## [1,]  NA   NA  NA  NA   NA  NA   NA  NA  NA   NA NA NA  NA   NA    NA
## [2,]  NA   NA  NA  NA   NA  NA   NA  NA  NA   NA NA NA  NA   NA    NA
## [3,]  NA   NA  NA  NA   NA  NA   NA  NA  NA   NA NA NA  NA   NA    NA
## [4,]  NA   NA  NA  NA   NA  NA   NA  NA  NA   NA NA NA  NA   NA    NA
## [5,]  NA   NA  NA  NA   NA  NA   NA  NA  NA   NA NA NA  NA   NA    NA
## [6,]  NA   NA  NA  NA   NA  NA   NA  NA  NA   NA NA NA  NA   NA    NA
##      attribute
## [1,]        NA
## [2,]        NA
## [3,]        NA
## [4,]        NA
## [5,]        NA
## [6,]        NA
```

```
onehrc<-na.omit(onehr)
sum(length(which(is.na(onehrc))))
```

```
## [1] 0
```

```
library(MVN)
class(onehrc)
```

```
## [1] "tbl_df"     "tbl"        "data.frame"
```

```
onehrc2<-onehrc[,-1]

lapply(onehrc2,class) #because the missing value is ? therefore R record is as character, we want it as
```

```
## $wsr0
## [1] "character"
##
## $wsr1
## [1] "character"
##
## $wsr2
## [1] "character"
##
## $wsr3
## [1] "character"
##
## $wsr4
## [1] "character"
##
## $wsr5
## [1] "character"
##
## $wsr6
## [1] "character"
##
```

```
## $wsr7
## [1] "character"
##
## $wsr8
## [1] "character"
##
## $wsr9
## [1] "character"
##
## $wsr10
## [1] "character"
##
## $wsr11
## [1] "character"
##
## $wsr12
## [1] "character"
##
## $wsr13
## [1] "character"
##
## $wsr14
## [1] "character"
##
## $wsr15
## [1] "character"
##
## $wsr16
## [1] "character"
##
## $wsr17
## [1] "character"
##
## $wsr18
## [1] "character"
##
## $wsr19
## [1] "character"
##
## $wsr20
## [1] "character"
##
## $wsr21
## [1] "character"
##
## $wsr22
## [1] "character"
##
## $wsr23
## [1] "character"
##
## $wsr_pk
## [1] "character"
##
```

```
## $wsr_av
## [1] "character"
##
## $t0
## [1] "character"
##
## $t1
## [1] "character"
##
## $t2
## [1] "character"
##
## $t3
## [1] "character"
##
## $t4
## [1] "character"
##
## $t5
## [1] "character"
##
## $t6
## [1] "character"
##
## $t7
## [1] "character"
##
## $t8
## [1] "character"
##
## $t9
## [1] "character"
##
## $t10
## [1] "character"
##
## $t11
## [1] "character"
##
## $t12
## [1] "character"
##
## $t13
## [1] "character"
##
## $t14
## [1] "character"
##
## $t15
## [1] "character"
##
## $t16
## [1] "character"
##
```

```
## $t17
## [1] "character"
##
## $t18
## [1] "character"
##
## $t19
## [1] "character"
##
## $t20
## [1] "character"
##
## $t21
## [1] "character"
##
## $t22
## [1] "character"
##
## $t23
## [1] "character"
##
## $t_pk
## [1] "numeric"
##
## $t_av
## [1] "numeric"
##
## $t85
## [1] "character"
##
## $rh85
## [1] "character"
##
## $u85
## [1] "character"
##
## $v85
## [1] "character"
##
## $ht85
## [1] "character"
##
## $t70
## [1] "character"
##
## $rh70
## [1] "character"
##
## $u70
## [1] "character"
##
## $v70
## [1] "character"
##
```

```
## $ht70
## [1] "character"
##
## $t50
## [1] "character"
##
## $rh50
## [1] "character"
##
## $u50
## [1] "character"
##
## $v50
## [1] "character"
##
## $ht50
## [1] "character"
##
## $ki
## [1] "character"
##
## $tt
## [1] "character"
##
## $slp
## [1] "character"
##
## $slp_
## [1] "character"
##
## $percp
## [1] "numeric"
##
## $attribute
## [1] "integer"
```

```
write.table(onehrc2,file="C:/Users/Hafid Pradipta/OneDrive/Documentos/onehrc2.csv",sep=",",qmethod="doul
head(onehrc2)
```

```
## # A tibble: 6 x 73
##    wsr0  wsr1  wsr2  wsr3  wsr4  wsr5  wsr6  wsr7  wsr8  wsr9 wsr10 wsr11
##   <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr>
## 1   0.8   1.8   2.4   2.1     2   2.1   1.5   1.7   1.9   2.3   3.7   5.5
## 2   2.8   3.2   3.3   2.7   3.3   3.2   2.9   2.8   3.1   3.4   4.2   4.5
## 3   2.9   2.8   2.6   2.1   2.2   2.5   2.5   2.7   2.2   2.5   3.1     4
## 4   4.7   3.8   3.7   3.8   2.9   3.1   2.8   2.5   2.4   3.1   3.3   3.1
## 5   3.7   3.2   3.8   5.1     6     7   6.3   6.4   6.3   5.4   6.1   6.4
## 6   2.2   2.9   3.4   4.2   4.7   4.7   5.3   4.9   5.2     6   5.9   6.1
## # ... with 61 more variables: wsr12 <chr>, wsr13 <chr>, wsr14 <chr>,
## #   wsr15 <chr>, wsr16 <chr>, wsr17 <chr>, wsr18 <chr>, wsr19 <chr>,
## #   wsr20 <chr>, wsr21 <chr>, wsr22 <chr>, wsr23 <chr>, wsr_pk <chr>,
## #   wsr_av <chr>, t0 <chr>, t1 <chr>, t2 <chr>, t3 <chr>, t4 <chr>,
## #   t5 <chr>, t6 <chr>, t7 <chr>, t8 <chr>, t9 <chr>, t10 <chr>,
## #   t11 <chr>, t12 <chr>, t13 <chr>, t14 <chr>, t15 <chr>, t16 <chr>,
## #   t17 <chr>, t18 <chr>, t19 <chr>, t20 <chr>, t21 <chr>, t22 <chr>,
```

```
## #   t23 <chr>, t_pk <dbl>, t_av <dbl>, t85 <chr>, rh85 <chr>, u85 <chr>,
## #   v85 <chr>, ht85 <chr>, t70 <chr>, rh70 <chr>, u70 <chr>, v70 <chr>,
## #   ht70 <chr>, t50 <chr>, rh50 <chr>, u50 <chr>, v50 <chr>, ht50 <chr>,
## #   ki <chr>, tt <chr>, slp <chr>, slp_ <chr>, percp <dbl>,
## #   attribute <int>
```

only take the average of the temperature and the windspeed

```
library(readr)
onehrc2 <- read_csv("C:/Users/Hafid Pradipta/OneDrive/STAT 520/dataset/Ozone/onehrc2.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_double()
## )
```

```
## See spec(...) for full column specifications.
```

Descriptive statistics

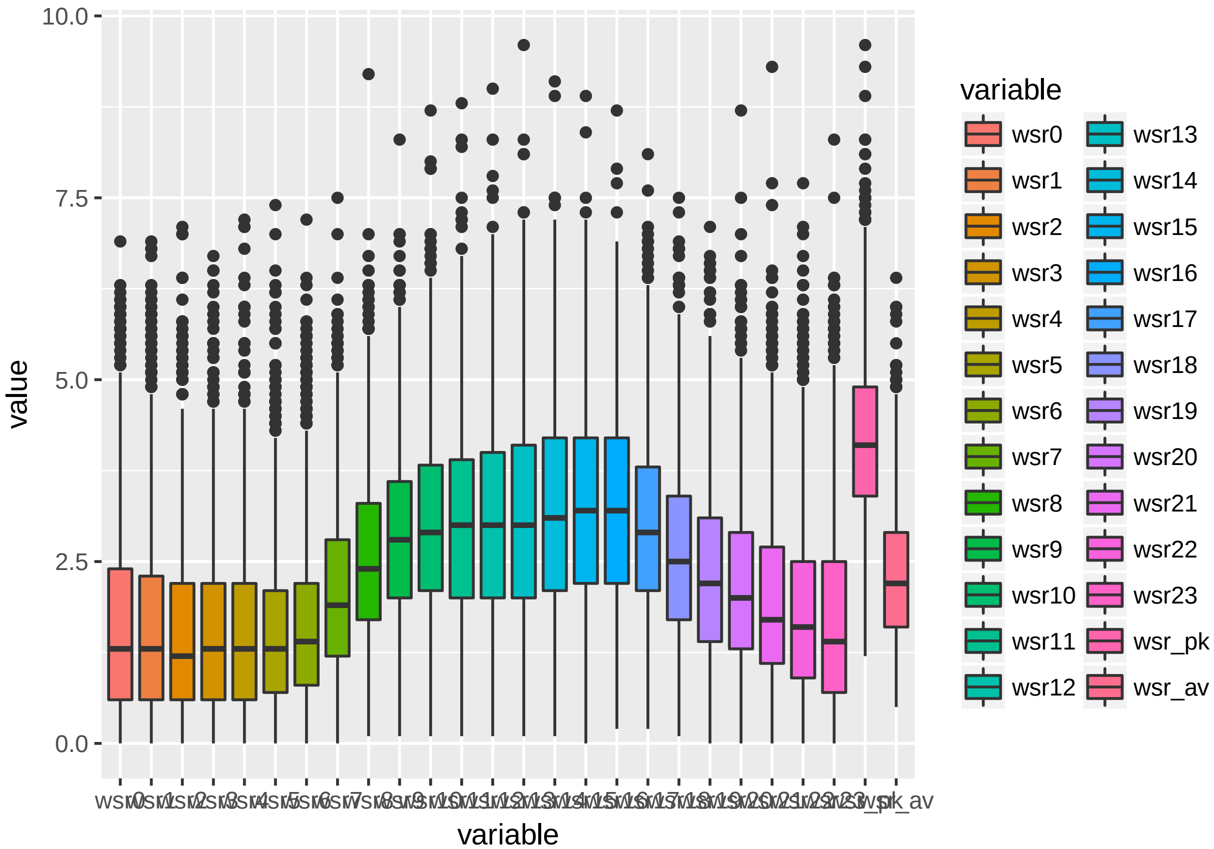
```
colnames(onehrc2)
```

```
##  [1] "wsr0"      "wsr1"      "wsr2"      "wsr3"      "wsr4"
##  [6] "wsr5"      "wsr6"      "wsr7"      "wsr8"      "wsr9"
## [11] "wsr10"     "wsr11"     "wsr12"     "wsr13"     "wsr14"
## [16] "wsr15"     "wsr16"     "wsr17"     "wsr18"     "wsr19"
## [21] "wsr20"     "wsr21"     "wsr22"     "wsr23"     "wsr_pk"
## [26] "wsr_av"    "t0"        "t1"        "t2"        "t3"
## [31] "t4"        "t5"        "t6"        "t7"        "t8"
## [36] "t9"        "t10"       "t11"       "t12"       "t13"
## [41] "t14"       "t15"       "t16"       "t17"       "t18"
## [46] "t19"       "t20"       "t21"       "t22"       "t23"
## [51] "t_pk"      "t_av"      "t85"       "rh85"      "u85"
## [56] "v85"       "ht85"      "t70"       "rh70"      "u70"
## [61] "v70"       "ht70"      "t50"       "rh50"      "u50"
## [66] "v50"       "ht50"      "ki"        "tt"        "slp"
## [71] "slp_"      "percp"     "attribute"
```

```
#windspeede at given time
wsx<-onehrc2[,1:26]
datawsx<-melt(wsx)
```

```
## No id variables; using all as measure variables
```

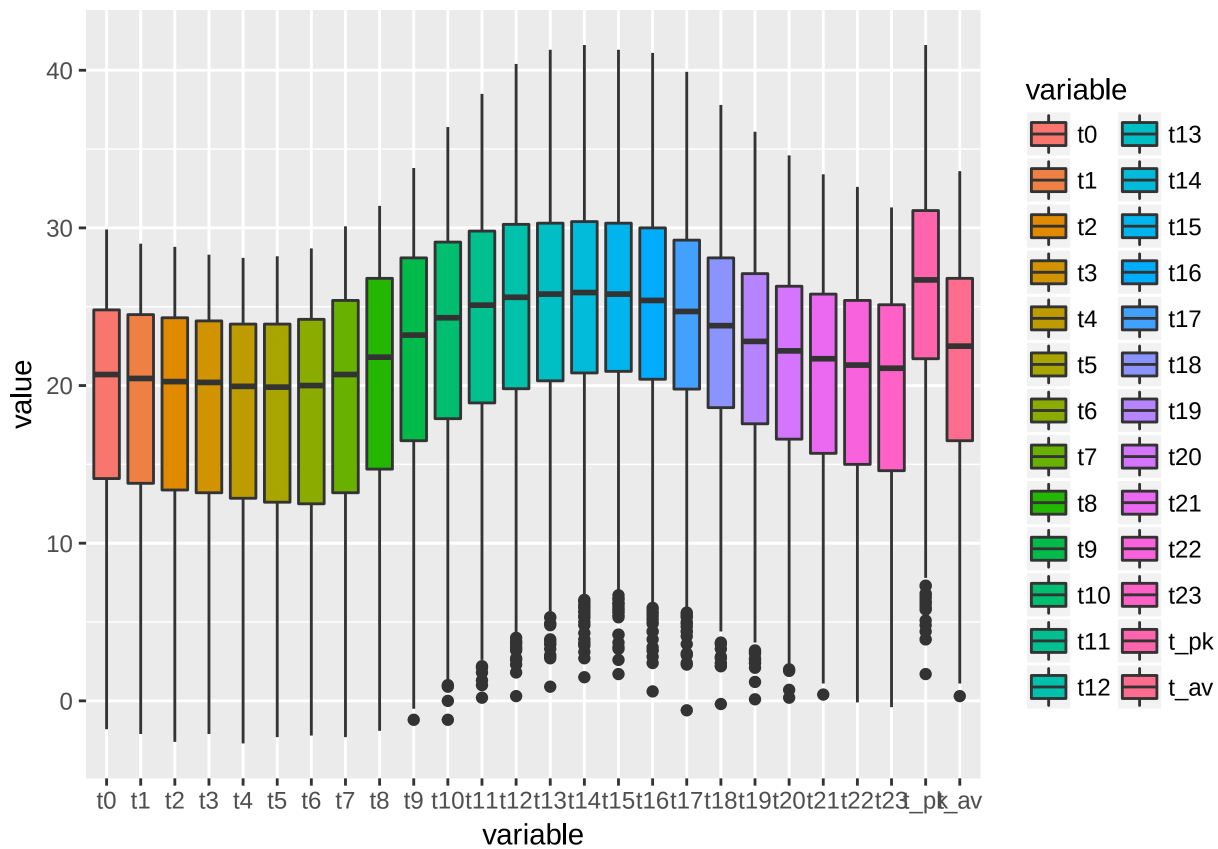
```
ggplot(datawsx,aes(x=variable, y=value, fill=variable)) + geom_boxplot()
```




```
#temperature at given time
tx<-onehrc2[,27:52]
datatx<-melt(tx)
```

```
## No id variables; using all as measure variables
```

```
ggplot(datatx,aes(x=variable, y=value, fill=variable)) + geom_boxplot()
```




```r
#grouping
#using average temperature and winspeed
names(onehrc2)
```

```
##  [1] "wsr0"      "wsr1"      "wsr2"      "wsr3"      "wsr4"
##  [6] "wsr5"      "wsr6"      "wsr7"      "wsr8"      "wsr9"
## [11] "wsr10"     "wsr11"     "wsr12"     "wsr13"     "wsr14"
## [16] "wsr15"     "wsr16"     "wsr17"     "wsr18"     "wsr19"
## [21] "wsr20"     "wsr21"     "wsr22"     "wsr23"     "wsr_pk"
## [26] "wsr_av"    "t0"        "t1"        "t2"        "t3"
## [31] "t4"        "t5"        "t6"        "t7"        "t8"
## [36] "t9"        "t10"       "t11"       "t12"       "t13"
## [41] "t14"       "t15"       "t16"       "t17"       "t18"
## [46] "t19"       "t20"       "t21"       "t22"       "t23"
## [51] "t_pk"      "t_av"      "t85"       "rh85"      "u85"
## [56] "v85"       "ht85"      "t70"       "rh70"      "u70"
## [61] "v70"       "ht70"      "t50"       "rh50"      "u50"
## [66] "v50"       "ht50"      "ki"        "tt"        "slp"
## [71] "slp_"      "percp"     "attribute"
```

```r
oneav<-onehrc2[,c(25,26,51:73)]


#check normality of whole dataset
uniNorm(oneav, type="SW")
```

```
## $`Descriptive Statistics`
```

```
##               n      Mean Std.Dev  Median    Min     Max      25th
## wsr_pk     1848     4.177   1.174     4.1    1.2     9.6     3.400
## wsr_av     1848     2.317   0.919     2.2    0.5     6.4     1.600
## t_pk       1848    25.892   6.859    26.7    1.7    41.6    21.700
## t_av       1848    21.164   6.749    22.5    0.3    33.6    16.500
## t85        1848    13.718   4.762    14.4   -4.5    24.5    10.800
## rh85       1848     0.582   0.256     0.6    0.0     1.0     0.400
## u85        1848     1.979   4.518     1.8  -15.8    18.3    -1.100
## v85        1848     1.944   6.094     1.7  -16.1    22.2    -2.200
## ht85       1848  1533.693  35.314  1537.0 1357.0  1642.0  1513.500
## t70        1848     6.077   3.801     6.8   -9.9    16.2     3.700
## rh70       1848     0.402   0.264     0.4    0.0     1.0     0.200
## u70        1848     5.167   6.329     4.8  -14.4    28.2     0.600
## v70        1848     1.012   6.294     0.9  -23.7    25.5    -2.800
## ht70       1848  3148.377  46.645  3156.0 2919.0  3249.0  3121.000
## t50        1848   -10.499   3.805   -10.1  -24.8    -1.7   -13.200
## rh50       1848     0.305   0.248     0.2    0.0     1.0     0.100
## u50        1848     9.820   9.345     9.2  -14.9    41.4     2.700
## v50        1848     0.647   7.354     0.3  -26.0    30.4    -4.000
## ht50       1848  5822.459  75.705  5835.0 5480.0  5965.0  5775.000
## ki         1848    10.690  20.175    14.7  -56.7    42.1    -2.725
## tt         1848    37.698  11.012    41.4  -10.1    59.2    33.000
## slp        1848 10165.446  52.058 10160.0 9995.0 10350.0 10130.000
## slp_       1848    -0.850  34.130     0.0 -135.0   140.0   -20.000
## percp      1848     0.360   1.263     0.0    0.0    20.7     0.000
## attribute  1848     0.031   0.173     0.0    0.0     1.0     0.000
##                 75th   Skew Kurtosis
## wsr_pk         4.900  0.537    0.559
## wsr_av         2.900  0.744    0.378
## t_pk          31.100 -0.643    0.009
## t_av          26.800 -0.618   -0.493
## t85           17.400 -0.667    0.092
## rh85           0.800 -0.483   -0.788
## u85            4.800  0.155    0.357
## v85            6.100  0.059   -0.093
## ht85        1557.500 -0.677    1.317
## t70            9.000 -0.753    0.476
## rh70           0.600  0.309   -0.967
## u70            9.800  0.155   -0.310
## v70            4.700  0.052    0.507
## ht70        3181.500 -0.852    1.086
## t50           -7.400 -0.423   -0.467
## rh50           0.500  0.797   -0.295
## u50           16.500  0.165   -0.495
## v50            4.625  0.313    0.695
## ht50        5880.000 -0.831    0.525
## ki            27.825 -0.778   -0.122
## tt            45.200 -1.297    1.423
## slp        10195.000  0.506    0.582
## slp_          15.000  0.297    1.819
## percp          0.100  7.230   76.489
## attribute      0.000  5.423   27.420
##
## $`Shapiro-Wilk's Normality Test`
```

```
##     Variable Statistic   p-value Normality
## 1   wsr_pk      0.9835    0.0000    NO
## 2   wsr_av      0.9600    0.0000    NO
## 3    t_pk       0.9676    0.0000    NO
## 4    t_av       0.9478    0.0000    NO
## 5     t85       0.9658    0.0000    NO
## 6    rh85       0.9375    0.0000    NO
## 7     u85       0.9947    0.0000    NO
## 8     v85       0.9988    0.2081    YES
## 9    ht85       0.9763    0.0000    NO
## 10    t70       0.9621    0.0000    NO
## 11   rh70       0.9458    0.0000    NO
## 12    u70       0.9954    0.0000    NO
## 13    v70       0.9954    0.0000    NO
## 14   ht70       0.9614    0.0000    NO
## 15    t50       0.9751    0.0000    NO
## 16   rh50       0.9018    0.0000    NO
## 17    u50       0.9933    0.0000    NO
## 18    v50       0.9900    0.0000    NO
## 19   ht50       0.9520    0.0000    NO
## 20    ki        0.9314    0.0000    NO
## 21    tt        0.8857    0.0000    NO
## 22    slp       0.9804    0.0000    NO
## 23   slp_       0.9702    0.0000    NO
## 24   percp      0.3099    0.0000    NO
## 25 attribute    0.1621    0.0000    NO
```

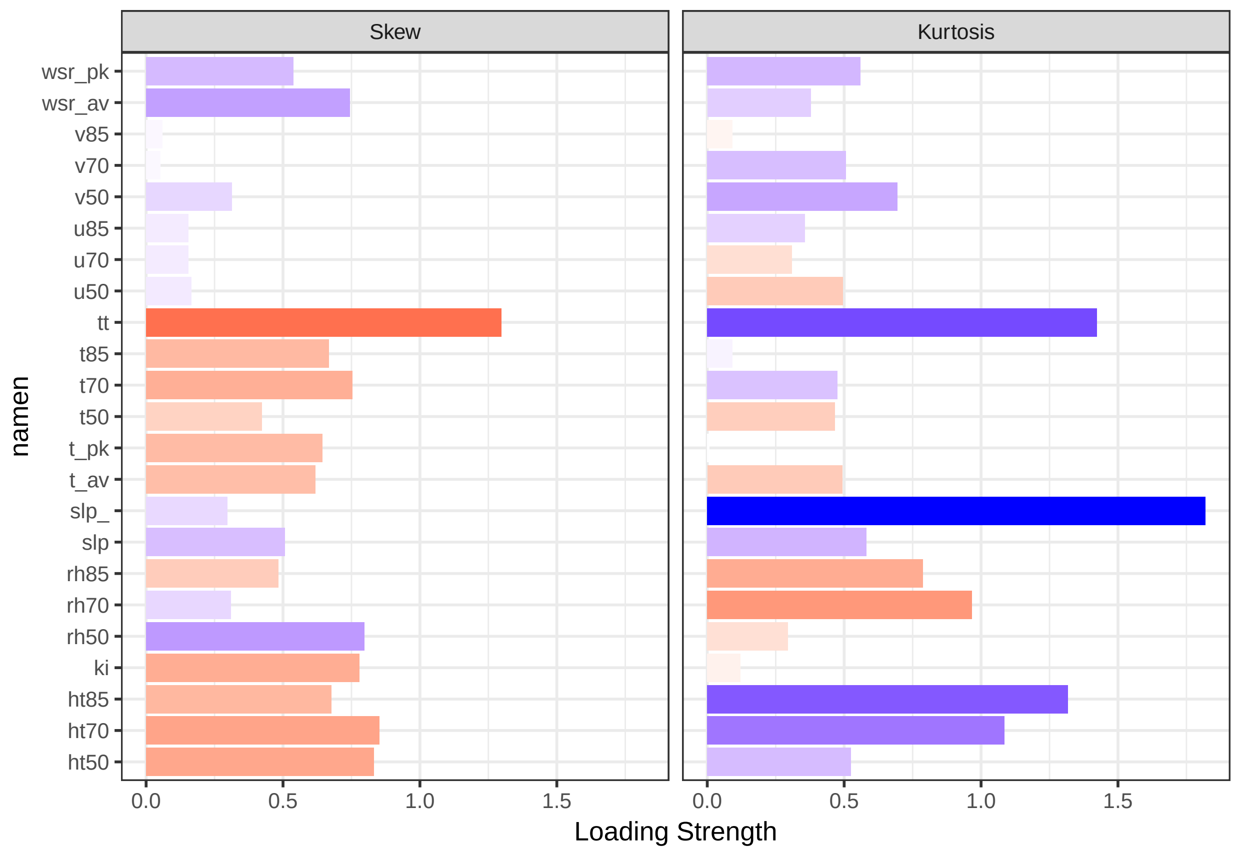
```
normtest<-uniNorm(oneav, type="SW")
normtest
```

```
## $`Descriptive Statistics`
##                n      Mean Std.Dev  Median    Min     Max      25th
## wsr_pk      1848     4.177   1.174     4.1    1.2     9.6     3.400
## wsr_av      1848     2.317   0.919     2.2    0.5     6.4     1.600
## t_pk        1848    25.892   6.859    26.7    1.7    41.6    21.700
## t_av        1848    21.164   6.749    22.5    0.3    33.6    16.500
## t85         1848    13.718   4.762    14.4   -4.5    24.5    10.800
## rh85        1848     0.582   0.256     0.6    0.0     1.0     0.400
## u85         1848     1.979   4.518     1.8  -15.8    18.3    -1.100
## v85         1848     1.944   6.094     1.7  -16.1    22.2    -2.200
## ht85        1848  1533.693  35.314  1537.0 1357.0  1642.0  1513.500
## t70         1848     6.077   3.801     6.8   -9.9    16.2     3.700
## rh70        1848     0.402   0.264     0.4    0.0     1.0     0.200
## u70         1848     5.167   6.329     4.8  -14.4    28.2     0.600
## v70         1848     1.012   6.294     0.9  -23.7    25.5    -2.800
## ht70        1848  3148.377  46.645  3156.0 2919.0  3249.0  3121.000
## t50         1848   -10.499   3.805   -10.1  -24.8    -1.7   -13.200
## rh50        1848     0.305   0.248     0.2    0.0     1.0     0.100
## u50         1848     9.820   9.345     9.2  -14.9    41.4     2.700
## v50         1848     0.647   7.354     0.3  -26.0    30.4    -4.000
## ht50        1848  5822.459  75.705  5835.0 5480.0  5965.0  5775.000
## ki          1848    10.690  20.175    14.7  -56.7    42.1    -2.725
## tt          1848    37.698  11.012    41.4  -10.1    59.2    33.000
## slp         1848 10165.446  52.058 10160.0 9995.0 10350.0 10130.000
## slp_        1848    -0.850  34.130     0.0 -135.0   140.0   -20.000
```

```
## percp     1848     0.360   1.263     0.0    0.0    20.7     0.000
## attribute 1848     0.031   0.173     0.0    0.0     1.0     0.000
##                75th   Skew Kurtosis
## wsr_pk        4.900  0.537    0.559
## wsr_av        2.900  0.744    0.378
## t_pk         31.100 -0.643    0.009
## t_av         26.800 -0.618   -0.493
## t85          17.400 -0.667    0.092
## rh85          0.800 -0.483   -0.788
## u85           4.800  0.155    0.357
## v85           6.100  0.059   -0.093
## ht85       1557.500 -0.677    1.317
## t70           9.000 -0.753    0.476
## rh70          0.600  0.309   -0.967
## u70           9.800  0.155   -0.310
## v70           4.700  0.052    0.507
## ht70       3181.500 -0.852    1.086
## t50          -7.400 -0.423   -0.467
## rh50          0.500  0.797   -0.295
## u50          16.500  0.165   -0.495
## v50           4.625  0.313    0.695
## ht50       5880.000 -0.831    0.525
## ki           27.825 -0.778   -0.122
## tt           45.200 -1.297    1.423
## slp       10195.000  0.506    0.582
## slp_         15.000  0.297    1.819
## percp         0.100  7.230   76.489
## attribute     0.000  5.423   27.420
##
## $`Shapiro-Wilk's Normality Test`
##     Variable Statistic   p-value Normality
## 1   wsr_pk      0.9835    0.0000    NO
## 2   wsr_av      0.9600    0.0000    NO
## 3    t_pk       0.9676    0.0000    NO
## 4    t_av       0.9478    0.0000    NO
## 5     t85       0.9658    0.0000    NO
## 6    rh85       0.9375    0.0000    NO
## 7     u85       0.9947    0.0000    NO
## 8     v85       0.9988    0.2081    YES
## 9    ht85       0.9763    0.0000    NO
## 10    t70       0.9621    0.0000    NO
## 11   rh70       0.9458    0.0000    NO
## 12    u70       0.9954    0.0000    NO
## 13    v70       0.9954    0.0000    NO
## 14   ht70       0.9614    0.0000    NO
## 15    t50       0.9751    0.0000    NO
## 16   rh50       0.9018    0.0000    NO
## 17    u50       0.9933    0.0000    NO
## 18    v50       0.9900    0.0000    NO
## 19   ht50       0.9520    0.0000    NO
## 20    ki        0.9314    0.0000    NO
## 21    tt        0.8857    0.0000    NO
## 22    slp       0.9804    0.0000    NO
## 23   slp_       0.9702    0.0000    NO
```

```
## 24    percp      0.3099    0.0000    NO
## 25 attribute     0.1621    0.0000    NO
```

```
#visualizatoin of normality dataset
name<-colnames(oneav)
namen<-colnames(oneav[1:23])
normplot<-normtest$`Descriptive Statistics`[1:23,9:10]
datanorm<-data.frame(namen,normplot)
loadings.1 <- melt(datanorm, id="namen",
                   measure=c("Skew","Kurtosis"),
                   variable.name="Factor", value.name="Loading")
ggplot(loadings.1, aes(namen, abs(Loading), fill=Loading)) +
  facet_wrap(~ Factor, nrow=1) + #place the factors in separate facets
  geom_bar(stat="identity") + #make the bars
  coord_flip() + #flip the axes so the test names can be horizontal
  #define the fill color gradient: blue=positive, red=negative
  scale_fill_gradient2(name = "Loading",
                       high = "blue", mid = "white", low = "red",
                       midpoint=0, guide=F) +
  ylab("Loading Strength") + #improve y-axis label
  theme_bw(base_size=10) #use a black-andOwhite theme with set font size
```




```
colnames(oneav)
```

```
##  [1] "wsr_pk"    "wsr_av"    "t_pk"      "t_av"      "t85"
##  [6] "rh85"      "u85"       "v85"       "ht85"      "t70"
## [11] "rh70"      "u70"       "v70"       "ht70"      "t50"
```

```
## [16] "rh50"      "u50"       "v50"       "ht50"      "ki"
## [21] "tt"        "slp"       "slp_"      "percp"     "attribute"
```

```r
#windspeedrate
wsr<-oneav[,1:2]
datawsr<- melt(wsr)
```

```
## No id variables; using all as measure variables
```

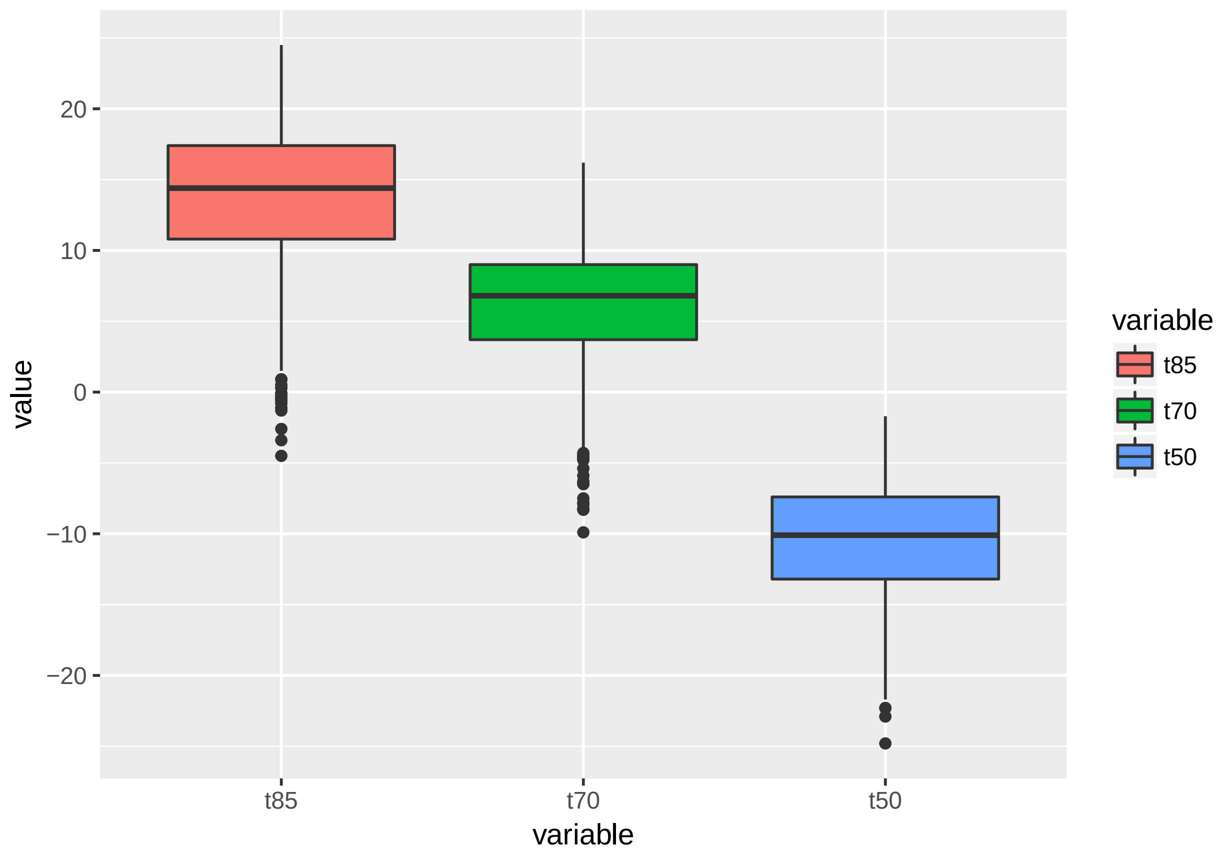
```r
#ggplot(datawsr,aes(x=variable, y=value, fill=variable)) + geom_boxplot()

#temperature
tempav<-oneav[,3:4]
datatempav<- melt(tempav)
```

```
## No id variables; using all as measure variables
```

```r
#ggplot(datatempav,aes(x=variable, y=value, fill=variable)) + geom_boxplot()


#various variable at 85hpa(1500m)
x85<-oneav[,5:9]

#various variable at 70hpa(3100m)
x70<-oneav[,10:14]

#various variable at 50hpa(5500m)
x50<-oneav[,15:19]

#temperature at various level
temp<-oneav[,c(5,10,15)]
datatemp<- melt(temp)
```

```
## No id variables; using all as measure variables
```

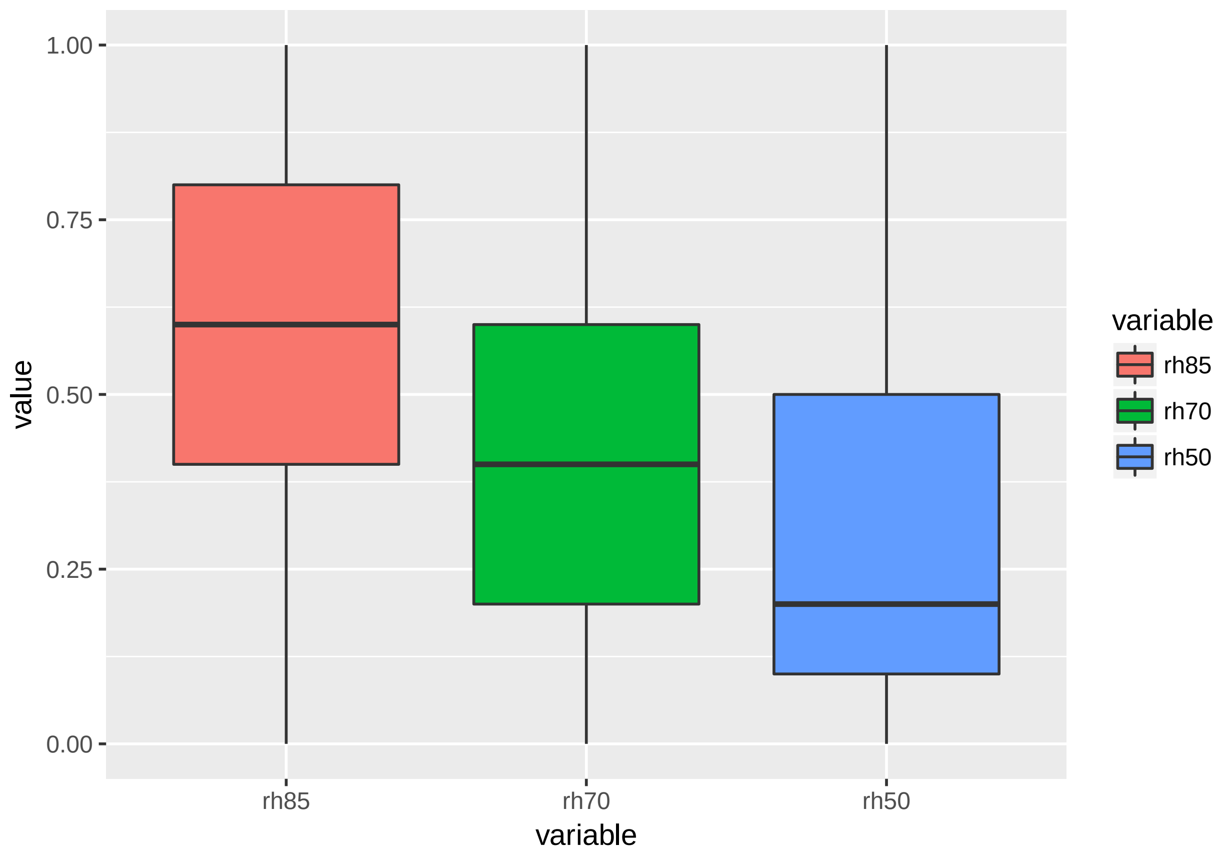
```r
ggplot(datatemp,aes(x=variable, y=value, fill=variable)) + geom_boxplot()
```




```r
#skew to the right, transformation for lambda<1

#relative humidity
rhum<-oneav[,c(6,11,16)]
datarhum<- melt(rhum)
```

```
## No id variables; using all as measure variables
```

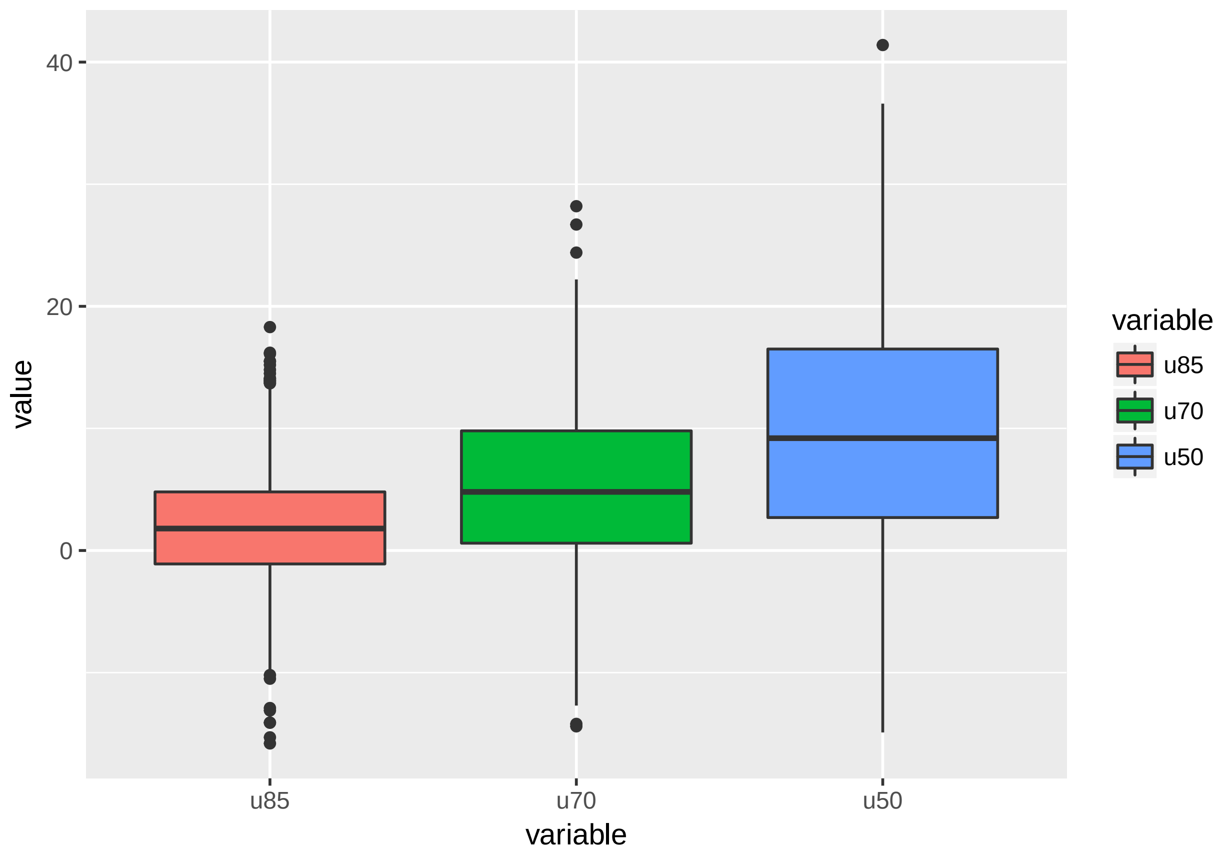
```r
ggplot(datarhum,aes(x=variable, y=value, fill=variable)) + geom_boxplot()
```




```
#skewed to the right

#Uwind
ns<-oneav[,c(7,12,17)]
datans<- melt(ns)
```

```
## No id variables; using all as measure variables
```

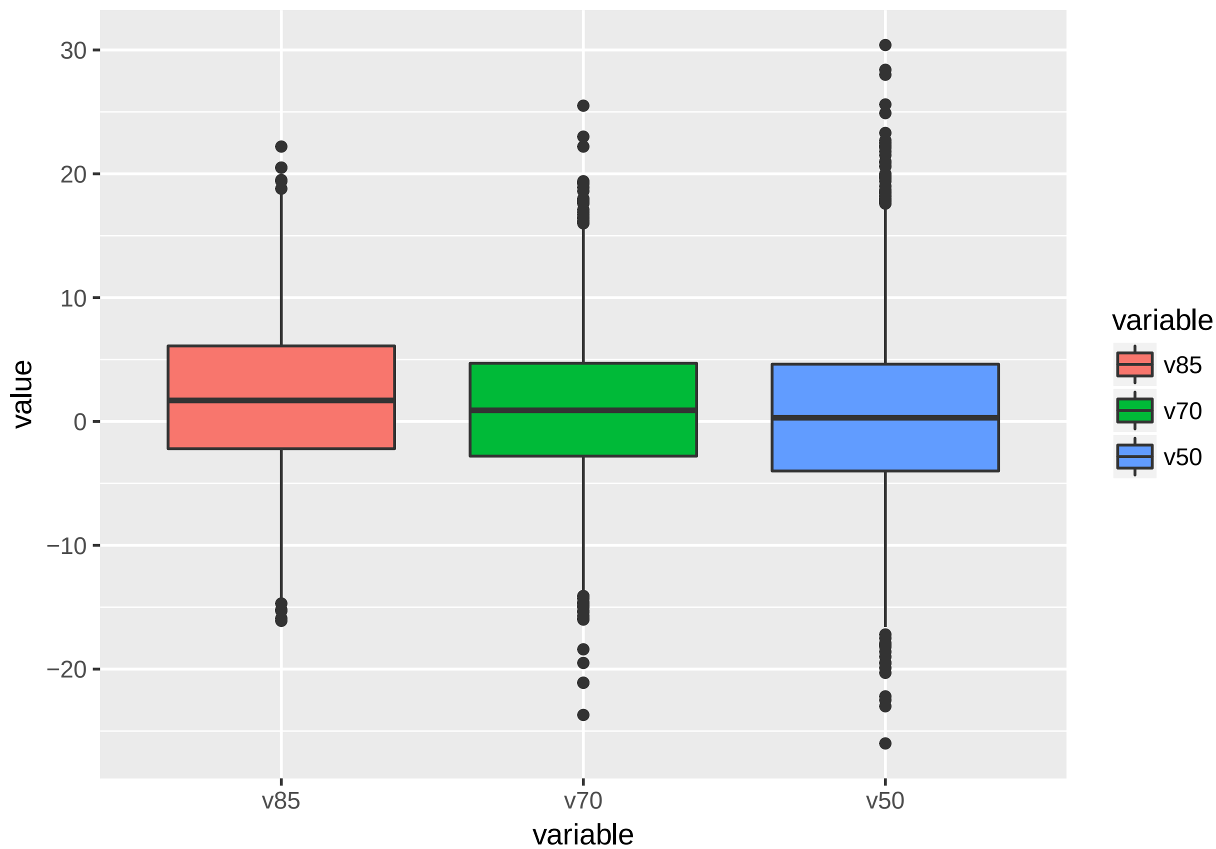
```
ggplot(datans,aes(x=variable, y=value, fill=variable)) + geom_boxplot()
```




```
#Vwind
ew<-oneav[,c(8,13,18)]
dataew<-melt(ew)
```

```
## No id variables; using all as measure variables
```

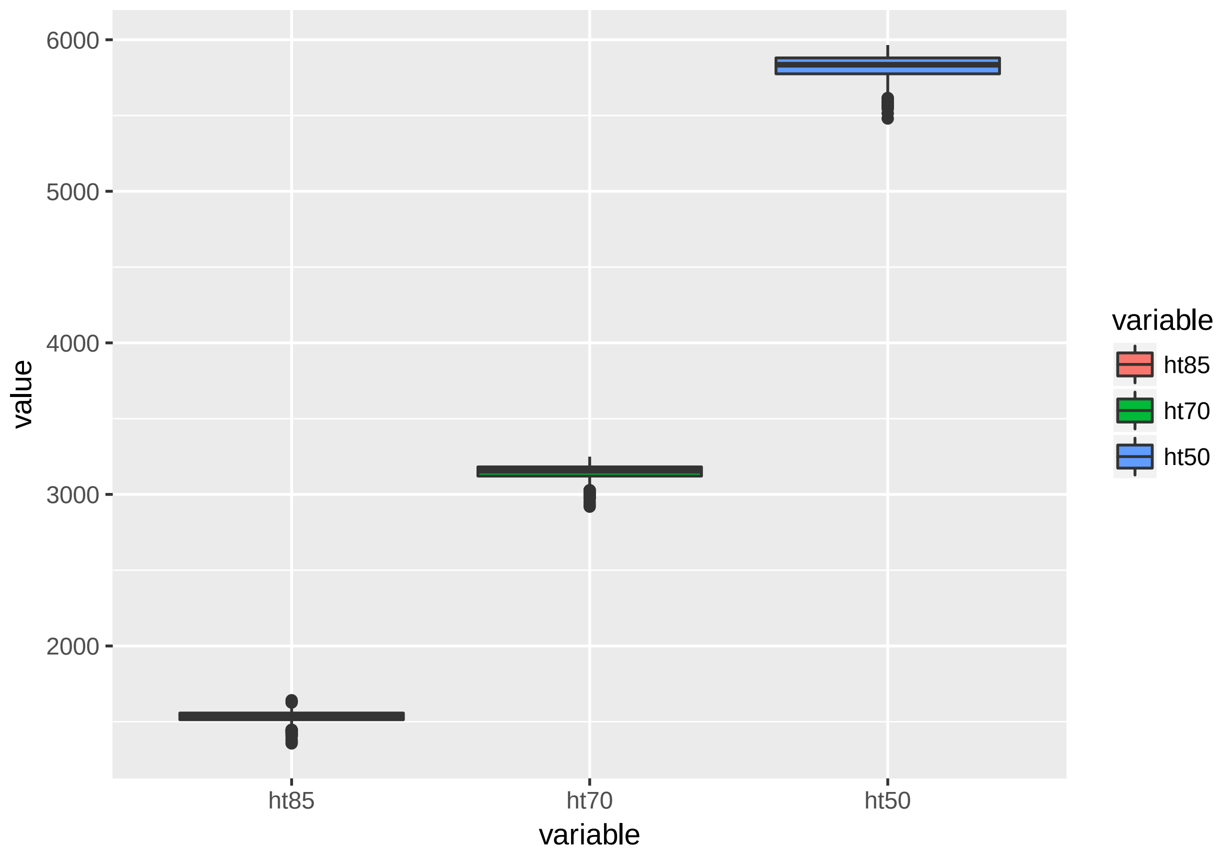
```
ggplot(dataew,aes(x=variable, y=value, fill=variable)) + geom_boxplot()
```



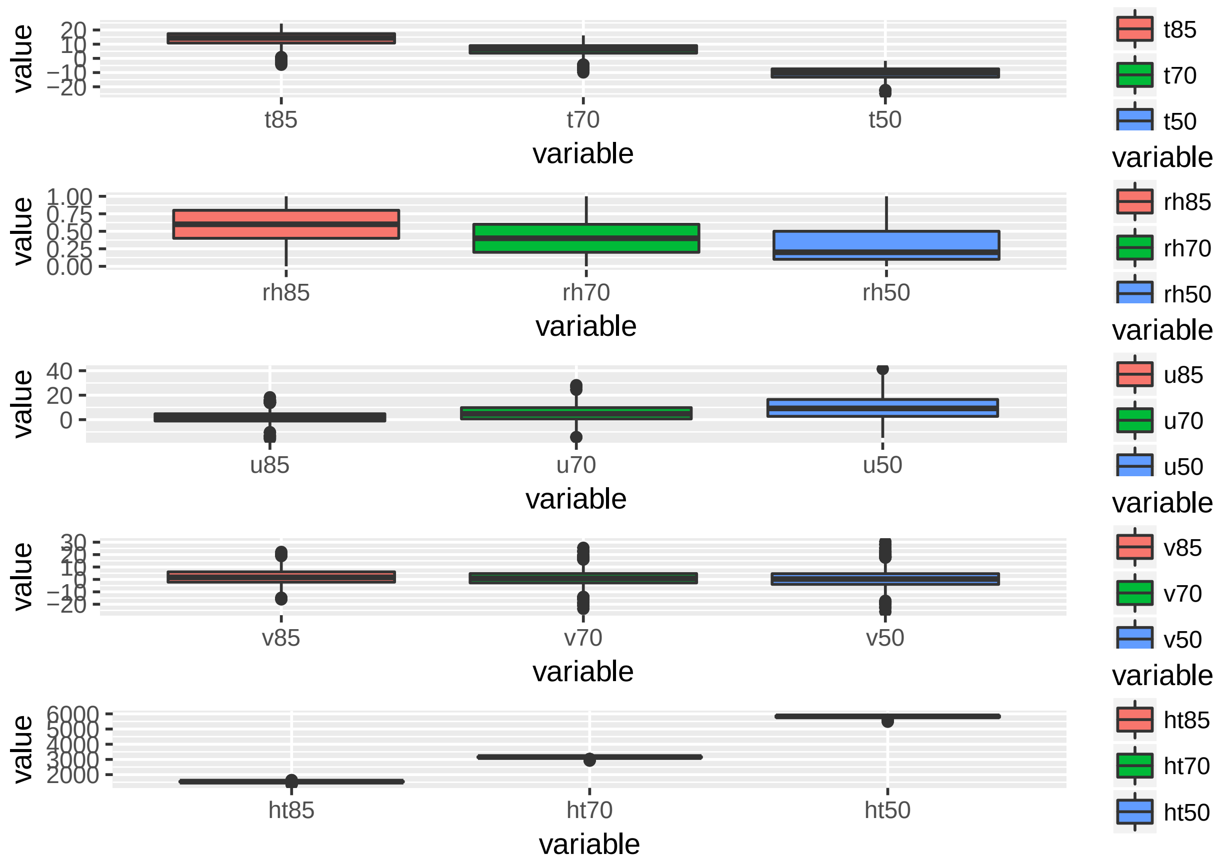

```r
#geopotential height
ht<-oneav[,c(9,14,19)]
dataht<-melt(ht)
```

```
## No id variables; using all as measure variables
```

```r
ggplot(dataht,aes(x=variable, y=value, fill=variable)) + geom_boxplot()
```



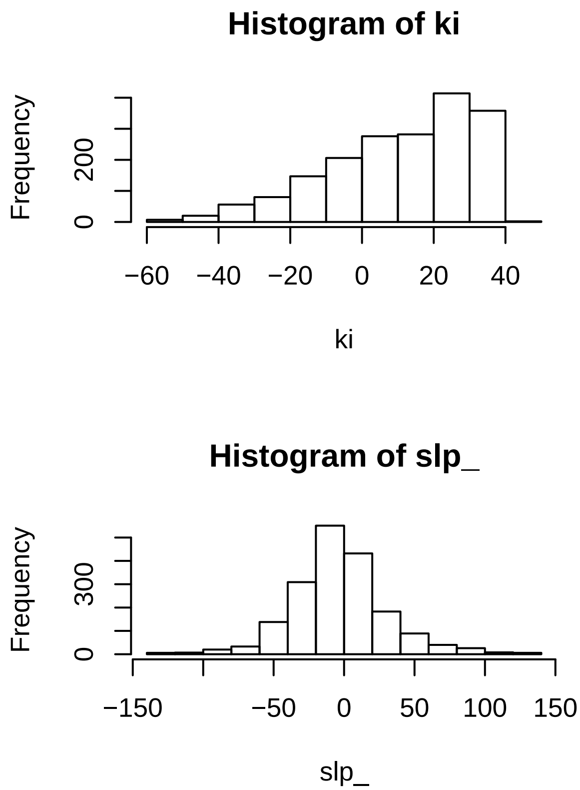

```
#combination of all plot
#install.packages("devtools")
library(devtools)
#install_github("easyGgplot2", "kassambara")
library(easyGgplot2)
plot1<-ggplot(datatemp,aes(x=variable, y=value, fill=variable)) + geom_boxplot()
plot2<-ggplot(datarhum,aes(x=variable, y=value, fill=variable)) + geom_boxplot()
plot3<-ggplot(datans,aes(x=variable, y=value, fill=variable)) + geom_boxplot()
plot4<-ggplot(dataew,aes(x=variable, y=value, fill=variable)) + geom_boxplot()
plot5<-ggplot(dataht,aes(x=variable, y=value, fill=variable)) + geom_boxplot()
ggplot2.multiplot(plot1,plot2,plot3,plot4,plot5, cols=1)
```



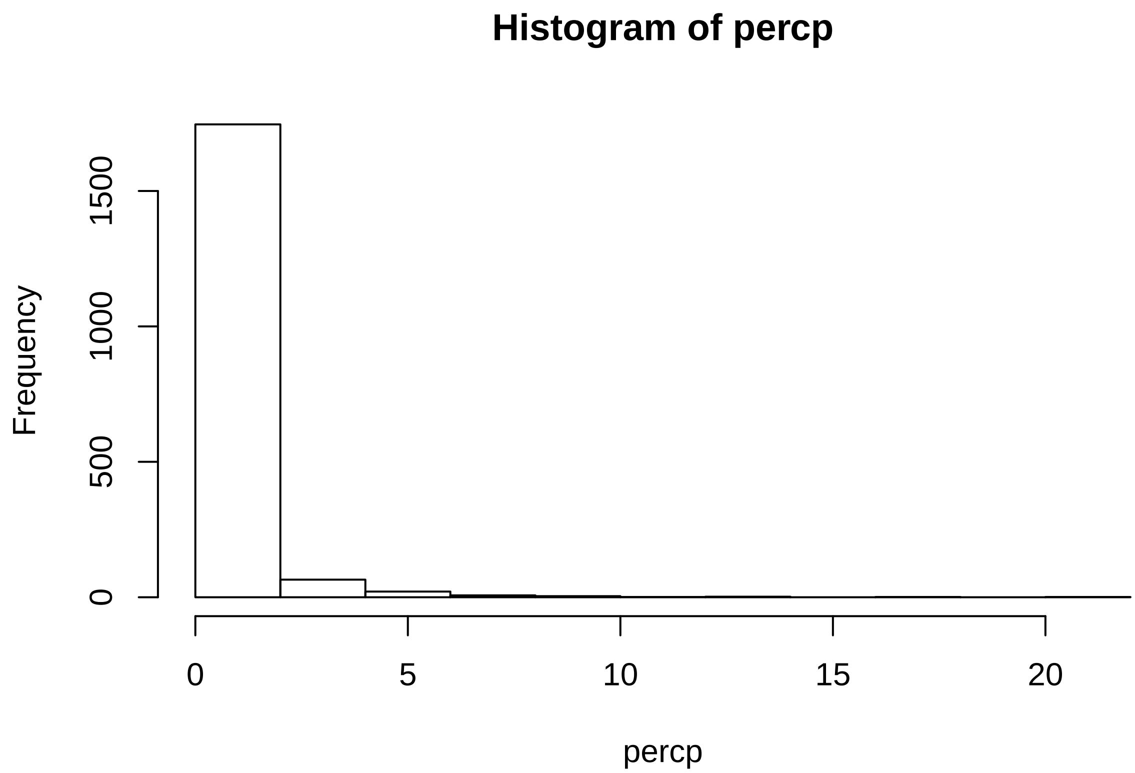

```r
names(oneav)
```

```
##  [1] "wsr_pk"    "wsr_av"    "t_pk"      "t_av"      "t85"
##  [6] "rh85"      "u85"       "v85"       "ht85"      "t70"
## [11] "rh70"      "u70"       "v70"       "ht70"      "t50"
## [16] "rh50"      "u50"       "v50"       "ht50"      "ki"
## [21] "tt"        "slp"       "slp_"      "percp"     "attribute"
```

```r
attach(oneav)
par(mfrow=c(2,2))
hist(ki)
hist(tt)
hist(slp_)
hist(slp)
```




```
par(mfrow=c(1,1))
hist(percp)
```



something to point out

```r
summary(oneav)
```

```
##      wsr_pk          wsr_av          t_pk            t_av
##  Min.   :1.200   Min.   :0.500   Min.   : 1.70   Min.   : 0.30
##  1st Qu.:3.400   1st Qu.:1.600   1st Qu.:21.70   1st Qu.:16.50
##  Median :4.100   Median :2.200   Median :26.70   Median :22.50
##  Mean   :4.177   Mean   :2.317   Mean   :25.89   Mean   :21.16
##  3rd Qu.:4.900   3rd Qu.:2.900   3rd Qu.:31.10   3rd Qu.:26.80
##  Max.   :9.600   Max.   :6.400   Max.   :41.60   Max.   :33.60
##       t85             rh85             u85               v85
##  Min.   :-4.50   Min.   :0.0000   Min.   :-15.800   Min.   :-16.100
##  1st Qu.:10.80   1st Qu.:0.4000   1st Qu.: -1.100   1st Qu.: -2.200
##  Median :14.40   Median :0.6000   Median :  1.800   Median :  1.700
##  Mean   :13.72   Mean   :0.5818   Mean   :  1.979   Mean   :  1.944
##  3rd Qu.:17.40   3rd Qu.:0.8000   3rd Qu.:  4.800   3rd Qu.:  6.100
##  Max.   :24.50   Max.   :1.0000   Max.   : 18.300   Max.   : 22.200
##       ht85           t70              rh70             u70
##  Min.   :1357   Min.   :-9.900   Min.   :0.000   Min.   :-14.400
##  1st Qu.:1514   1st Qu.: 3.700   1st Qu.:0.200   1st Qu.:  0.600
##  Median :1537   Median : 6.800   Median :0.400   Median :  4.800
##  Mean   :1534   Mean   : 6.077   Mean   :0.402   Mean   :  5.167
##  3rd Qu.:1558   3rd Qu.: 9.000   3rd Qu.:0.600   3rd Qu.:  9.800
```

```
##  Max.   :1642   Max.   :16.200   Max.   :1.000   Max.   : 28.200
##       v70               ht70           t50             rh50
##  Min.   :-23.700   Min.   :2919   Min.   :-24.8   Min.   :0.000
##  1st Qu.: -2.800   1st Qu.:3121   1st Qu.:-13.2   1st Qu.:0.100
##  Median :  0.900   Median :3156   Median :-10.1   Median :0.200
##  Mean   :  1.012   Mean   :3148   Mean   :-10.5   Mean   :0.305
##  3rd Qu.:  4.700   3rd Qu.:3182   3rd Qu.: -7.4   3rd Qu.:0.500
##  Max.   : 25.500   Max.   :3249   Max.   : -1.7   Max.   :1.000
##       u50               v50                 ht50            ki
##  Min.   :-14.90   Min.   :-26.0000   Min.   :5480   Min.   :-56.700
##  1st Qu.:  2.70   1st Qu.: -4.0000   1st Qu.:5775   1st Qu.: -2.725
##  Median :  9.20   Median :  0.3000   Median :5835   Median : 14.700
##  Mean   :  9.82   Mean   :  0.6467   Mean   :5822   Mean   : 10.690
##  3rd Qu.: 16.50   3rd Qu.:  4.6250   3rd Qu.:5880   3rd Qu.: 27.825
##  Max.   : 41.40   Max.   : 30.4000   Max.   :5965   Max.   : 42.100
##        tt              slp              slp_               percp
##  Min.   :-10.1   Min.   : 9995   Min.   :-135.0000   Min.   : 0.0000
##  1st Qu.: 33.0   1st Qu.:10130   1st Qu.: -20.0000   1st Qu.: 0.0000
##  Median : 41.4   Median :10160   Median :   0.0000   Median : 0.0000
##  Mean   : 37.7   Mean   :10165   Mean   :  -0.8496   Mean   : 0.3597
##  3rd Qu.: 45.2   3rd Qu.:10195   3rd Qu.:  15.0000   3rd Qu.: 0.1000
##  Max.   : 59.2   Max.   :10350   Max.   : 140.0000   Max.   :20.7000
##    attribute
##  Min.   :0.00000
##  1st Qu.:0.00000
##  Median :0.00000
##  Mean   :0.03084
##  3rd Qu.:0.00000
##  Max.   :1.00000
```

```
#the one that is not negative is relative humidity, geometric height, and slp
#skew to the left, try trasnformation for squared and cube
trrhum<-(rhum)^3
uniNorm(trrhum, type = "SW")
```

```
## $`Descriptive Statistics`
##         n  Mean Std.Dev Median Min Max  25th  75th  Skew Kurtosis
## rh85 1848 0.303   0.263  0.216   0   1 0.064 0.512 0.680   -0.410
## rh70 1848 0.155   0.215  0.064   0   1 0.008 0.216 1.834    3.202
## rh50 1848 0.097   0.177  0.008   0   1 0.001 0.125 2.558    6.678
##
## $`Shapiro-Wilk's Normality Test`
##    Variable Statistic   p-value Normality
## 1   rh85       0.8977         0    NO
## 2   rh70       0.7336         0    NO
## 3   rh50       0.6010         0    NO
```

```
summary(rhum)
```

```
##       rh85              rh70             rh50
##  Min.   :0.0000   Min.   :0.000   Min.   :0.000
##  1st Qu.:0.4000   1st Qu.:0.200   1st Qu.:0.100
##  Median :0.6000   Median :0.400   Median :0.200
##  Mean   :0.5818   Mean   :0.402   Mean   :0.305
##  3rd Qu.:0.8000   3rd Qu.:0.600   3rd Qu.:0.500
```

```
##  Max.   :1.0000   Max.   :1.000   Max.   :1.000
```

```r
#skew to the left, try transformation for squared and cube
trht<-ht^3
uniNorm(trht, type = "SW")
```

```
## $`Descriptive Statistics`
##         n         Mean    Std.Dev       Median          Min          Max
## ht85 1848   3613282500  245682940   3630961153   2498846293   4427101288
## ht70 1848  31228028371 1370317276  31434820416  24871517559  34296447249
## ht50 1848 197487086783 7618916469 198665557875 164566592000 212242007125
##              25th         75th   Skew Kurtosis
## ht85   3466947585   3778193234 -0.484    0.859
## ht70  30400540561  32202959268 -0.748    0.783
## ht50 192599859375 203297472000 -0.760    0.327
##
## $`Shapiro-Wilk's Normality Test`
##   Variable Statistic   p-value Normality
## 1   ht85       0.9863         0    NO
## 2   ht70       0.9691         0    NO
## 3   ht50       0.9579         0    NO
```

```r
#data contains negative
```

correlation Matrix

```r
#code taken from http://www.sthda.com/english/wiki/ggplot2-quick-correlation-matrix-heatmap-r-software-
#correlation matrix heatmap of the dataset
cormat<-round(cor(oneav),2)
cormat
```

```
##           wsr_pk wsr_av  t_pk  t_av   t85  rh85   u85   v85  ht85   t70
## wsr_pk      1.00   0.85 -0.17 -0.13 -0.09  0.03  0.18  0.39 -0.26 -0.09
## wsr_av      0.85   1.00 -0.23 -0.15 -0.08  0.06  0.18  0.48 -0.25 -0.04
## t_pk       -0.17  -0.23  1.00  0.97  0.83  0.17 -0.33  0.15  0.27  0.75
## t_av       -0.13  -0.15  0.97  1.00  0.86  0.27 -0.35  0.26  0.26  0.78
## t85        -0.09  -0.08  0.83  0.86  1.00  0.20 -0.27  0.32  0.19  0.87
## rh85        0.03   0.06  0.17  0.27  0.20  1.00 -0.04  0.32 -0.08  0.16
## u85         0.18   0.18 -0.33 -0.35 -0.27 -0.04  1.00  0.10 -0.42 -0.27
## v85         0.39   0.48  0.15  0.26  0.32  0.32  0.10  1.00 -0.12  0.33
## ht85       -0.26  -0.25  0.27  0.26  0.19 -0.08 -0.42 -0.12  1.00  0.19
## t70        -0.09  -0.04  0.75  0.78  0.87  0.16 -0.27  0.33  0.19  1.00
## rh70       -0.04  -0.02  0.07  0.15  0.22  0.52 -0.10  0.15 -0.02 -0.06
## u70         0.20   0.23 -0.61 -0.62 -0.54 -0.12  0.78 -0.01 -0.47 -0.50
## v70         0.34   0.41 -0.05  0.05  0.18  0.33  0.13  0.81 -0.21  0.19
## ht70       -0.25  -0.22  0.64  0.65  0.66  0.06 -0.46  0.09  0.85  0.65
## t50        -0.21  -0.18  0.72  0.76  0.81  0.24 -0.37  0.17  0.28  0.84
## rh50        0.03   0.05 -0.01  0.07  0.13  0.32 -0.02  0.22 -0.04  0.03
## u50         0.21   0.23 -0.68 -0.69 -0.62 -0.16  0.64 -0.04 -0.46 -0.56
## v50         0.26   0.29 -0.24 -0.15  0.01  0.28  0.18  0.54 -0.32  0.00
## ht50       -0.23  -0.20  0.75  0.78  0.81  0.14 -0.45  0.18  0.64  0.84
## ki         -0.01   0.01  0.41  0.50  0.54  0.75 -0.12  0.31 -0.01  0.28
## tt          0.07   0.09  0.39  0.47  0.47  0.81  0.00  0.38 -0.10  0.29
```

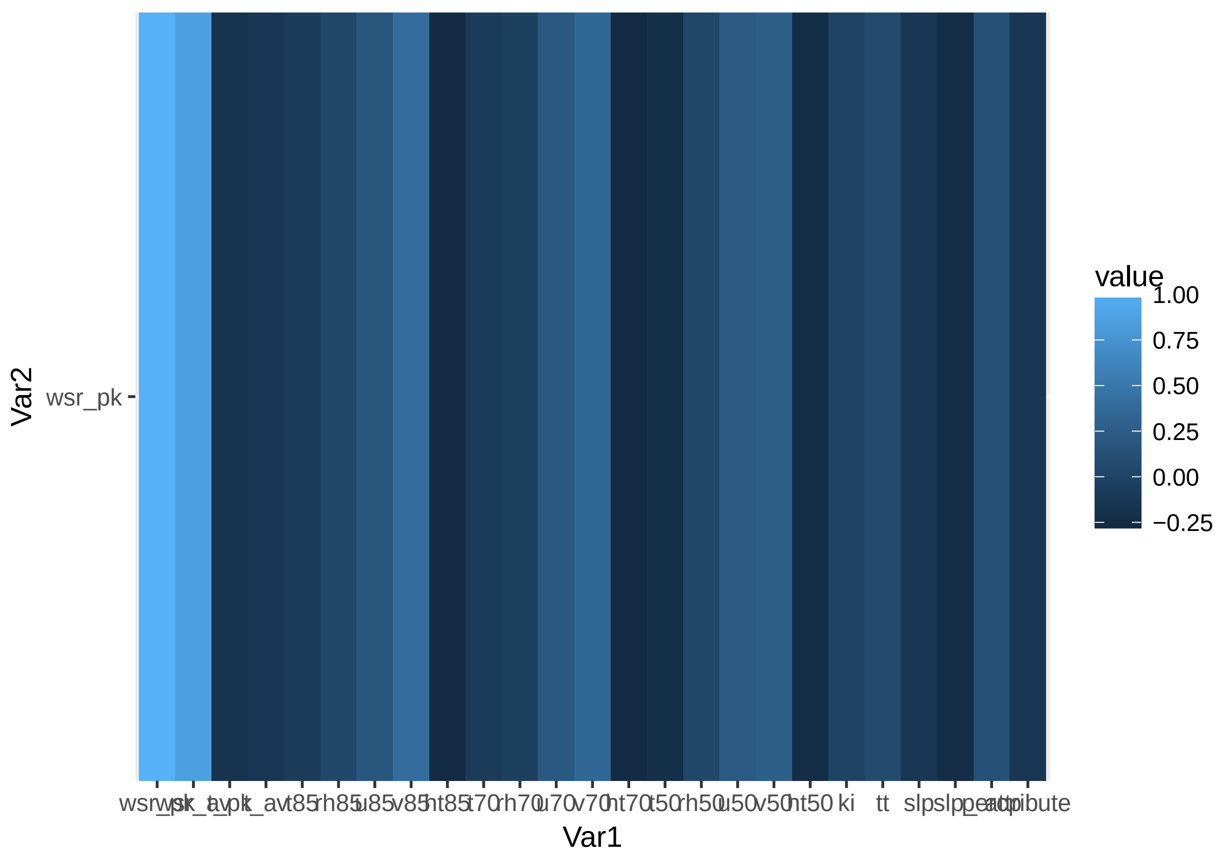
```
## slp        -0.13  -0.10 -0.47 -0.50 -0.58 -0.35 -0.09 -0.33  0.65 -0.48
## slp_       -0.23  -0.21 -0.14 -0.17 -0.13 -0.22 -0.09 -0.45  0.22 -0.10
## percp       0.12   0.07 -0.06 -0.01  0.05  0.17 -0.05  0.19 -0.05  0.00
## attribute  -0.13  -0.16  0.17  0.14  0.12 -0.02 -0.11 -0.09  0.07  0.09
##            rh70   u70   v70  ht70   t50  rh50   u50   v50  ht50    ki
## wsr_pk    -0.04  0.20  0.34 -0.25 -0.21  0.03  0.21  0.26 -0.23 -0.01
## wsr_av    -0.02  0.23  0.41 -0.22 -0.18  0.05  0.23  0.29 -0.20  0.01
## t_pk       0.07 -0.61 -0.05  0.64  0.72 -0.01 -0.68 -0.24  0.75  0.41
## t_av       0.15 -0.62  0.05  0.65  0.76  0.07 -0.69 -0.15  0.78  0.50
## t85        0.22 -0.54  0.18  0.66  0.81  0.13 -0.62  0.01  0.81  0.54
## rh85       0.52 -0.12  0.33  0.06  0.24  0.32 -0.16  0.28  0.14  0.75
## u85       -0.10  0.78  0.13 -0.46 -0.37 -0.02  0.64  0.18 -0.45 -0.12
## v85        0.15 -0.01  0.81  0.09  0.17  0.22 -0.04  0.54  0.18  0.31
## ht85      -0.02 -0.47 -0.21  0.85  0.28 -0.04 -0.46 -0.32  0.64 -0.01
## t70       -0.06 -0.50  0.19  0.65  0.84  0.03 -0.56  0.00  0.84  0.28
## rh70       1.00 -0.10  0.23  0.07  0.19  0.51 -0.16  0.25  0.09  0.80
## u70       -0.10  1.00  0.12 -0.65 -0.59  0.00  0.88  0.24 -0.67 -0.24
## v70        0.23  0.12  1.00 -0.05  0.10  0.28  0.09  0.79  0.04  0.29
## ht70       0.07 -0.65 -0.05  1.00  0.67  0.04 -0.67 -0.23  0.94  0.25
## t50        0.19 -0.59  0.10  0.67  1.00  0.02 -0.62 -0.02  0.86  0.39
## rh50       0.51  0.00  0.28  0.04  0.02  1.00 -0.01  0.30  0.04  0.45
## u50       -0.16  0.88  0.09 -0.67 -0.62 -0.01  1.00  0.25 -0.70 -0.32
## v50        0.25  0.24  0.79 -0.23 -0.02  0.30  0.25  1.00 -0.15  0.23
## ht50       0.09 -0.67  0.04  0.94  0.86  0.04 -0.70 -0.15  1.00  0.33
## ki         0.80 -0.24  0.29  0.25  0.39  0.45 -0.32  0.23  0.33  1.00
## tt         0.47 -0.16  0.32  0.15  0.21  0.34 -0.24  0.22  0.22  0.85
## slp       -0.21  0.08 -0.29  0.19 -0.38 -0.14  0.15 -0.25 -0.09 -0.48
## slp_      -0.13  0.05 -0.32  0.10 -0.04 -0.11  0.08 -0.19  0.03 -0.20
## percp      0.25 -0.01  0.24 -0.02  0.03  0.23  0.00  0.21  0.00  0.20
## attribute -0.01 -0.15 -0.09  0.11  0.11 -0.07 -0.14 -0.09  0.12  0.03
##              tt   slp  slp_ percp attribute
## wsr_pk     0.07 -0.13 -0.23  0.12     -0.13
## wsr_av     0.09 -0.10 -0.21  0.07     -0.16
## t_pk       0.39 -0.47 -0.14 -0.06      0.17
## t_av       0.47 -0.50 -0.17 -0.01      0.14
## t85        0.47 -0.58 -0.13  0.05      0.12
## rh85       0.81 -0.35 -0.22  0.17     -0.02
## u85        0.00 -0.09 -0.09 -0.05     -0.11
## v85        0.38 -0.33 -0.45  0.19     -0.09
## ht85      -0.10  0.65  0.22 -0.05      0.07
## t70        0.29 -0.48 -0.10  0.00      0.09
## rh70       0.47 -0.21 -0.13  0.25     -0.01
## u70       -0.16  0.08  0.05 -0.01     -0.15
## v70        0.32 -0.29 -0.32  0.24     -0.09
## ht70       0.15  0.19  0.10 -0.02      0.11
## t50        0.21 -0.38 -0.04  0.03      0.11
## rh50       0.34 -0.14 -0.11  0.23     -0.07
## u50       -0.24  0.15  0.08  0.00     -0.14
## v50        0.22 -0.25 -0.19  0.21     -0.09
## ht50       0.22 -0.09  0.03  0.00      0.12
## ki         0.85 -0.48 -0.20  0.20      0.03
## tt         1.00 -0.53 -0.24  0.14      0.02
## slp       -0.53  1.00  0.32 -0.09     -0.05
## slp_      -0.24  0.32  1.00 -0.08     -0.01
```

```
## percp      0.14 -0.09 -0.08  1.00      -0.04
## attribute  0.02 -0.05 -0.01 -0.04       1.00
```

```
#reshape
library(reshape2)
melted_cormat<-melt(cormat)
head(melted_cormat)
```

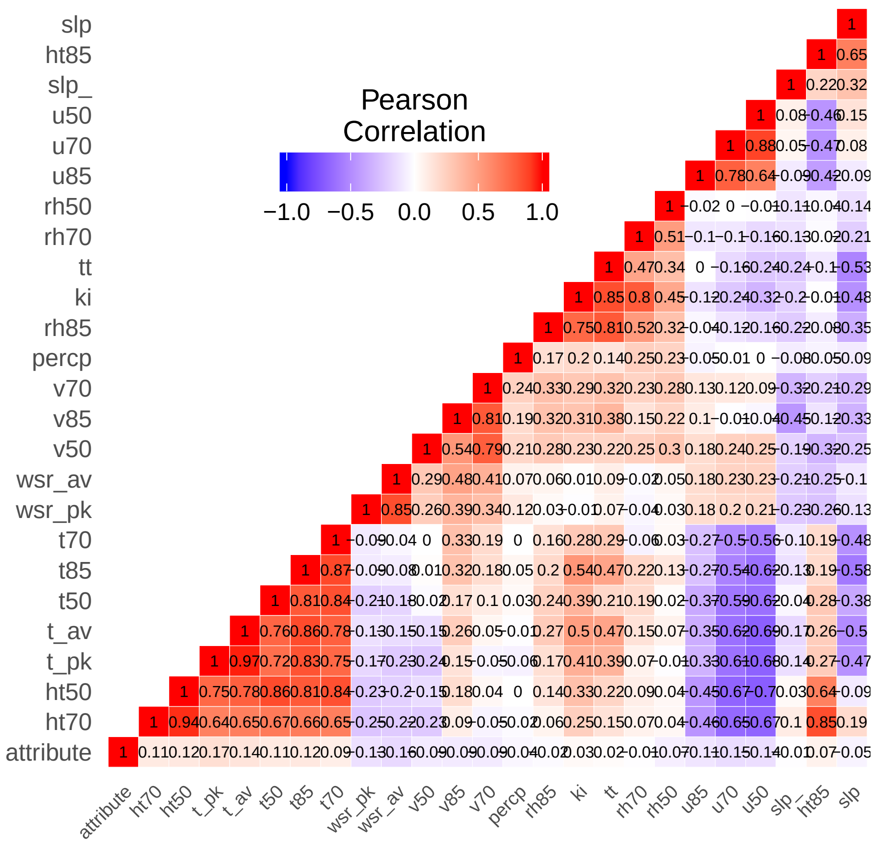
```
##     Var1   Var2 value
## 1 wsr_pk wsr_pk  1.00
## 2 wsr_av wsr_pk  0.85
## 3   t_pk wsr_pk -0.17
## 4   t_av wsr_pk -0.13
## 5    t85 wsr_pk -0.09
## 6   rh85 wsr_pk  0.03
```

```
attach(melted_cormat)
melted_cormat<-melted_cormat[Var1,]
library(ggplot2)
ggplot(data=melted_cormat, aes(x=Var1, y=Var2, fill=value))+geom_tile()
```




```
get_lower_tri<-function(cormat){
  cormat[upper.tri(cormat)] <- NA
  return(cormat)
}
# Get upper triangle of the correlation matrix
get_upper_tri <- function(cormat){
  cormat[lower.tri(cormat)]<- NA
```

```r
  return(cormat)
}

reorder_cormat <- function(cormat){
  # Use correlation between variables as distance
  dd <- as.dist((1-cormat)/2)
  hc <- hclust(dd)
  cormat <-cormat[hc$order, hc$order]
}

cormat <- reorder_cormat(cormat)
upper_tri <- get_upper_tri(cormat)
# Melt the correlation matrix
melted_cormat <- melt(upper_tri, na.rm = TRUE)
# Create a ggheatmap
ggheatmap <- ggplot(melted_cormat, aes(Var2, Var1, fill = value))+
  geom_tile(color = "white")+
  scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                       midpoint = 0, limit = c(-1,1), space = "Lab",
                       name="Pearson\nCorrelation") +
  theme_minimal()+ # minimal theme
  theme(axis.text.x = element_text(angle = 45, vjust = 1,
                                   size = 7, hjust = 1))+
  coord_fixed()

ggheatmap +
  geom_text(aes(Var2, Var1, label = value), color = "black", size = 2) +
  theme(
    axis.title.x = element_blank(),
    axis.title.y = element_blank(),
    panel.grid.major = element_blank(),
    panel.border = element_blank(),
    panel.background = element_blank(),
    axis.ticks = element_blank(),
    legend.justification = c(1, 0),
    legend.position = c(0.6, 0.7),
    legend.direction = "horizontal")+
  guides(fill = guide_colorbar(barwidth = 7, barheight = 1,
                               title.position = "top", title.hjust = 0.5))
```



Eigen values and variances explained

```r
fitpca <- princomp(oneav, cor=TRUE)
summary(fitpca) # print variance accounted for
```

```
## Importance of components:
##                           Comp.1    Comp.2     Comp.3     Comp.4    Comp.5
## Standard deviation     2.8475059 2.2186642 1.49690912 1.38280567 1.1165494
## Proportion of Variance 0.3243316 0.1968988 0.08962948 0.07648606 0.0498673
## Cumulative Proportion  0.3243316 0.5212304 0.61085990 0.68734596 0.7372133
##                            Comp.6     Comp.7     Comp.8     Comp.9
## Standard deviation     1.04173087 0.98076082 0.93256131 0.89296877
## Proportion of Variance 0.04340813 0.03847567 0.03478682 0.03189573
## Cumulative Proportion  0.78062139 0.81909706 0.85388388 0.88577961
##                           Comp.10   Comp.11    Comp.12    Comp.13
## Standard deviation     0.85342516 0.7249121 0.63323021 0.52187084
## Proportion of Variance 0.02913338 0.0210199 0.01603922 0.01089397
## Cumulative Proportion  0.91491299 0.9359329 0.95197211 0.96286608
##                           Comp.14     Comp.15     Comp.16     Comp.17
## Standard deviation     0.49703295 0.471209589 0.347798097 0.320645664
## Proportion of Variance 0.00988167 0.008881539 0.004838541 0.004112546
## Cumulative Proportion  0.97274775 0.981629290 0.986467830 0.990580376
##                            Comp.18     Comp.19     Comp.20      Comp.21
## Standard deviation     0.298215297 0.244025829 0.188064097 0.1311093615
```

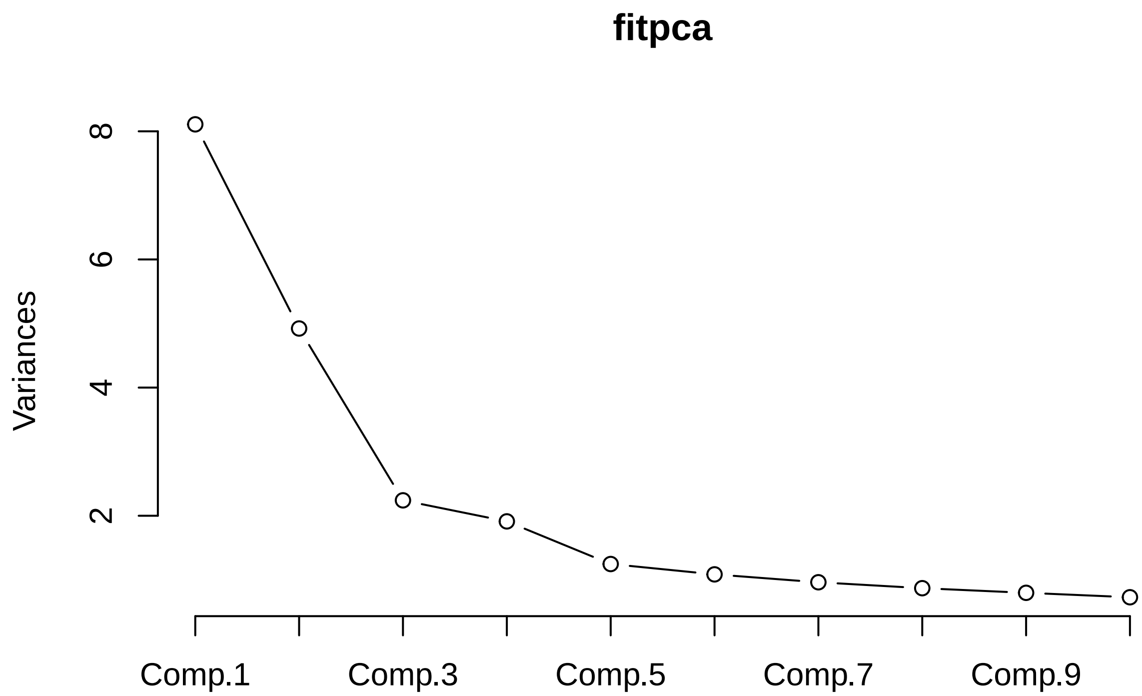
```
## Proportion of Variance 0.003557295 0.002381944 0.001414724 0.0006875866
## Cumulative Proportion  0.994137671 0.996519615 0.997934339 0.9986219256
##                             Comp.22      Comp.23      Comp.24      Comp.25
## Standard deviation     0.1278202305 0.1061695382 0.0665481177 4.912459e-02
## Proportion of Variance 0.0006535205 0.0004508788 0.0001771461 9.652901e-05
## Cumulative Proportion  0.9992754461 0.9997263249 0.9999034710 1.000000e+00
```

```r
loadings(fitpca) # pc loadings
```

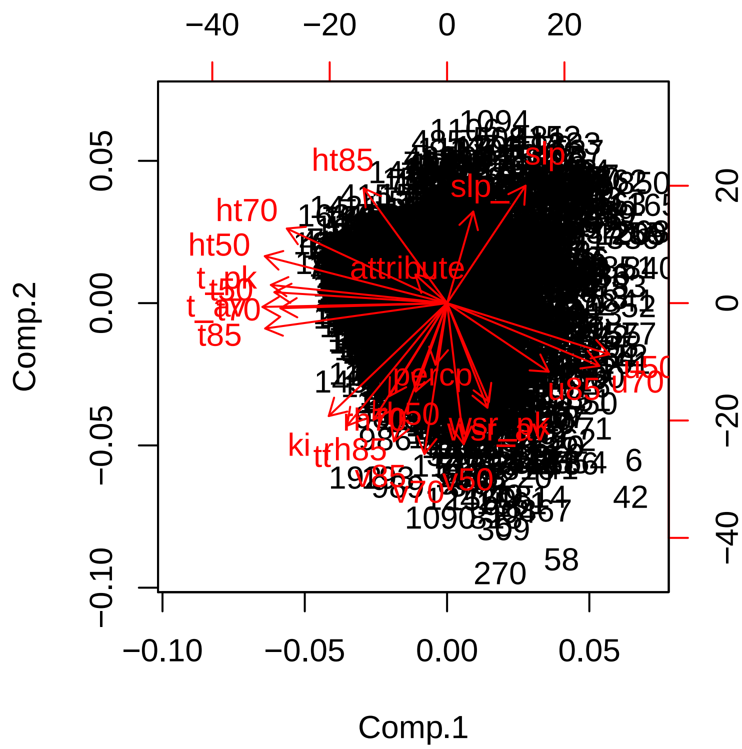
```
##
## Loadings:
##           Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8 Comp.9
## wsr_pk           -0.221 -0.329  0.177 -0.411 -0.198        -0.287       
## wsr_av           -0.234 -0.335  0.220 -0.335 -0.246        -0.241 -0.124
## t_pk      -0.307               -0.208                              0.115
## t_av      -0.321               -0.153                                   
## t85       -0.317        -0.119 -0.123  0.114               -0.116       
## rh85      -0.128 -0.261  0.298               -0.184 -0.155              
## u85        0.177 -0.153        -0.214  0.350 -0.365 -0.151         0.329
## v85              -0.309 -0.265  0.213               -0.164  0.161       
## ht85      -0.145  0.256         0.439        -0.215 -0.204         0.141
## t70       -0.290        -0.258         0.195                            
## rh70      -0.101 -0.208  0.428  0.112                                   
## u70        0.265 -0.140        -0.117  0.316 -0.252        -0.150  0.143
## v70              -0.337 -0.171  0.248  0.250  0.195 -0.113  0.123 -0.142
## ht70      -0.279  0.167         0.273  0.133 -0.180 -0.100              
## t50       -0.301                       0.181         0.128              
## rh50             -0.197  0.280  0.179                0.146              
## u50        0.282 -0.114                0.266 -0.156        -0.139       
## v50              -0.315         0.189  0.324  0.269               -0.318
## ht50      -0.317  0.105         0.137  0.173                            
## ki        -0.206 -0.252  0.331               -0.169                     
## tt        -0.174 -0.274  0.224        -0.116 -0.260 -0.149              
## slp        0.137  0.262         0.467        -0.133 -0.189              
## slp_              0.205  0.132         0.255 -0.193  0.310 -0.466 -0.532
## percp            -0.138  0.107  0.197         0.373  0.382 -0.396  0.599
## attribute                      -0.141         0.367 -0.697 -0.575       
##           Comp.10 Comp.11 Comp.12 Comp.13 Comp.14 Comp.15 Comp.16 Comp.17
## wsr_pk                             0.319           0.180   0.568  -0.177 
## wsr_av                    -0.135  -0.147          -0.128  -0.623   0.237 
## t_pk                       0.300   0.192   0.209   0.388  -0.113   0.115 
## t_av               0.102   0.236           0.205   0.348  -0.225   0.128 
## t85       -0.115           0.131                  -0.411                 
## rh85       0.411   0.134  -0.467                   0.329                 
## u85               -0.106           0.146  -0.595   0.146          -0.128 
## v85                0.113   0.263  -0.543           0.139          -0.477 
## ht85                               0.139                                 
## t70                0.101  -0.321  -0.100          -0.227                 
## rh70      -0.246  -0.582   0.170  -0.154                                 
## u70                                        0.253           0.108   0.415 
## v70                        0.179  -0.115                   0.300   0.563 
## ht70                                              -0.155                 
## t50               -0.347  -0.414                   0.272                 
## rh50      -0.694   0.505  -0.213                   0.114                 
## u50                       -0.104           0.674                  -0.280 
```

```
## v50                                0.623                    -0.294  -0.200
## ht50                      -0.177
## ki                -0.189   0.132                   -0.167
## tt         0.252   0.321   0.106   0.103           -0.353
## slp
## slp_       0.220   0.208   0.259  -0.160  -0.142    0.161
## percp      0.345
## attribute
##           Comp.18 Comp.19 Comp.20 Comp.21 Comp.22 Comp.23 Comp.24 Comp.25
## wsr_pk
## wsr_av
## t_pk               0.115          -0.424  -0.516  -0.117
## t_av                               0.412   0.518   0.316
## t85       -0.130  -0.159   0.510          -0.222   0.463  -0.107   0.176
## rh85               0.126   0.356   0.175  -0.216
## u85       -0.206   0.107
## v85        0.271
## ht85              -0.103   0.125           0.105  -0.361  -0.418   0.470
## t70        0.258   0.551  -0.330                          -0.326   0.119
## rh70               0.410   0.173  -0.190   0.175
## u70        0.578  -0.286   0.110
## v70       -0.414
## ht70                       0.143   0.154          -0.173          -0.796
## t50       -0.151  -0.473  -0.143  -0.304   0.190          -0.207  -0.113
## rh50
## u50       -0.437   0.204
## v50        0.211
## ht50                                              -0.158   0.812   0.289
## ki                -0.260  -0.561   0.389  -0.344
## tt                        -0.104  -0.499   0.385
## slp                       -0.264  -0.225  -0.117   0.685
## slp_
## percp
## attribute
##
##                Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8
## SS loadings      1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00
## Proportion Var   0.04   0.04   0.04   0.04   0.04   0.04   0.04   0.04
## Cumulative Var   0.04   0.08   0.12   0.16   0.20   0.24   0.28   0.32
##                Comp.9 Comp.10 Comp.11 Comp.12 Comp.13 Comp.14 Comp.15
## SS loadings      1.00    1.00    1.00    1.00    1.00    1.00    1.00
## Proportion Var   0.04    0.04    0.04    0.04    0.04    0.04    0.04
## Cumulative Var   0.36    0.40    0.44    0.48    0.52    0.56    0.60
##                Comp.16 Comp.17 Comp.18 Comp.19 Comp.20 Comp.21 Comp.22
## SS loadings       1.00    1.00    1.00    1.00    1.00    1.00    1.00
## Proportion Var    0.04    0.04    0.04    0.04    0.04    0.04    0.04
## Cumulative Var    0.64    0.68    0.72    0.76    0.80    0.84    0.88
##                Comp.23 Comp.24 Comp.25
## SS loadings       1.00    1.00    1.00
## Proportion Var    0.04    0.04    0.04
## Cumulative Var    0.92    0.96    1.00
```

```
plot(fitpca,type="lines") # scree plot
```

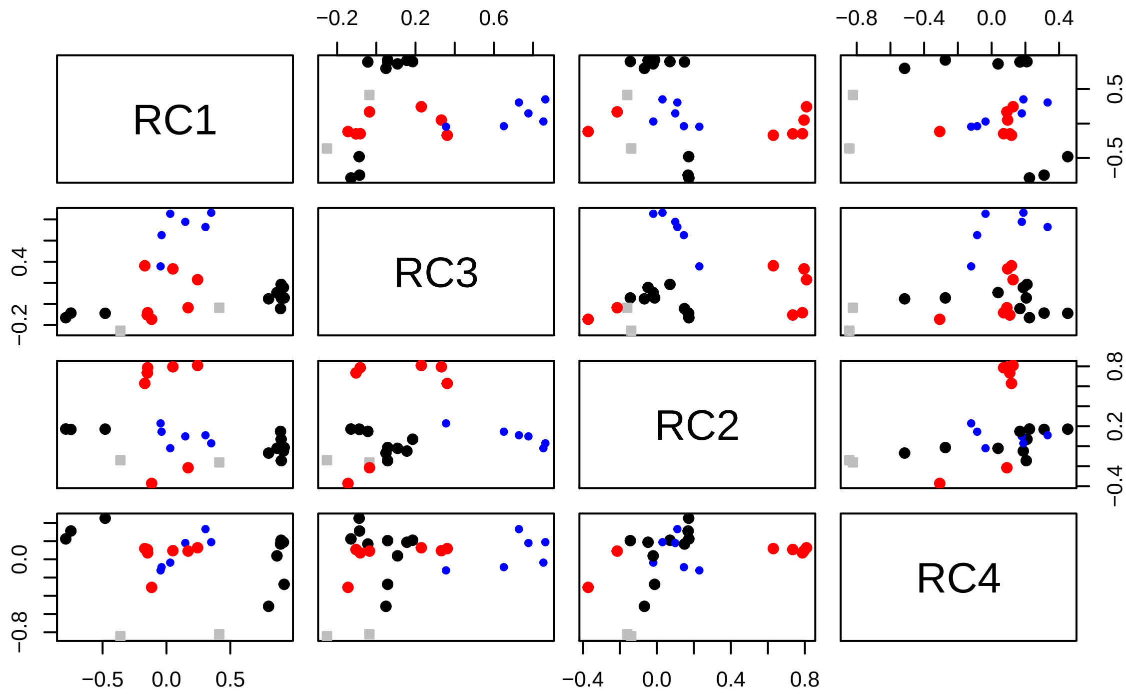



```r
biplot(fitpca)
```

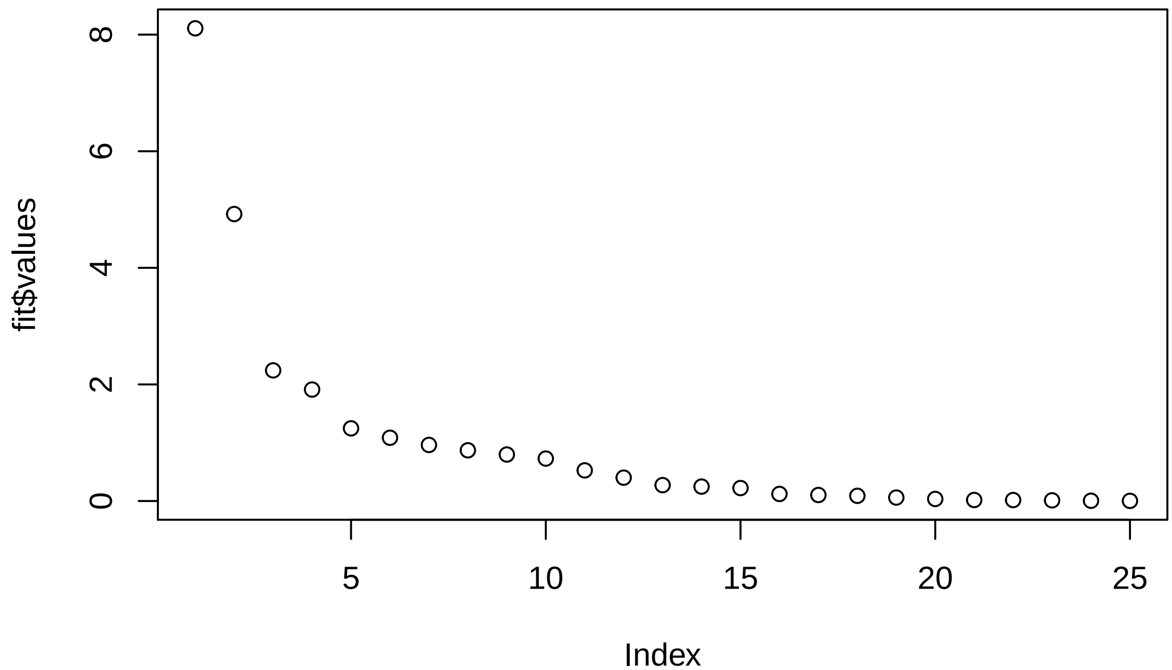



```
fit <- principal(oneav, nfactors=4, rotate="varimax")
plot(fit)
```

Principal Component Analysis




```r
plot(fit$values)
```




```
fit$values
```

```
##  [1] 8.108289780 4.922470805 2.240736904 1.912151516 1.246682488
##  [6] 1.085203212 0.961891781 0.869670594 0.797393229 0.728334505
## [11] 0.525497529 0.400980496 0.272349172 0.247041758 0.222038477
## [16] 0.120963516 0.102813642 0.088932363 0.059548605 0.035368105
## [21] 0.017189665 0.016338011 0.011271971 0.004428652 0.002413225
```

```
#total variances
sum(fit$values[1:4])/sum(fit$values)
```

```
## [1] 0.687346
```

```
names(fit)
```

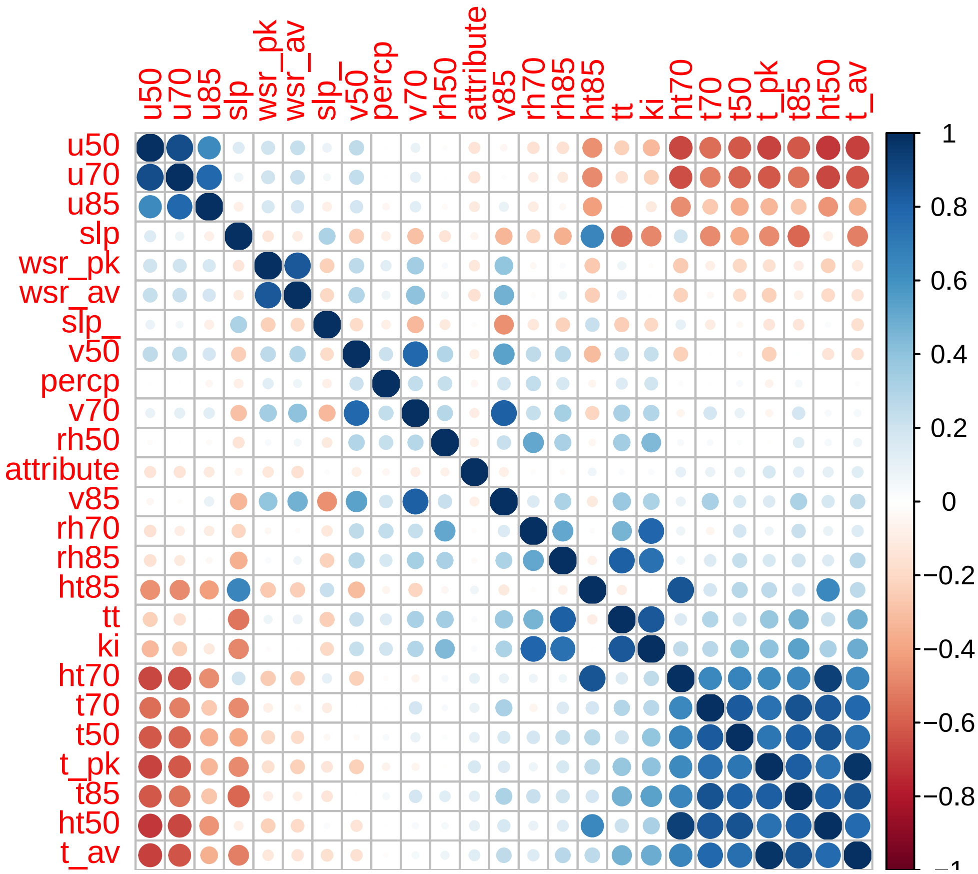
```
##  [1] "values"       "rotation"     "n.obs"        "communality"
##  [5] "loadings"     "fit"          "fit.off"      "fn"
##  [9] "Call"         "uniquenesses" "complexity"   "chi"
## [13] "EPVAL"        "R2"           "objective"    "residual"
## [17] "rms"          "factors"      "dof"          "null.dof"
## [21] "null.model"   "criteria"     "STATISTIC"    "PVAL"
## [25] "weights"      "r.scores"     "rot.mat"      "Vaccounted"
## [29] "Structure"    "scores"
```

```
name<-colnames(oneav)
fitloading<-data.frame(fit$loadings[,1:4],name)
fitloading
```

```
##                    RC1         RC3         RC2         RC4       name
```

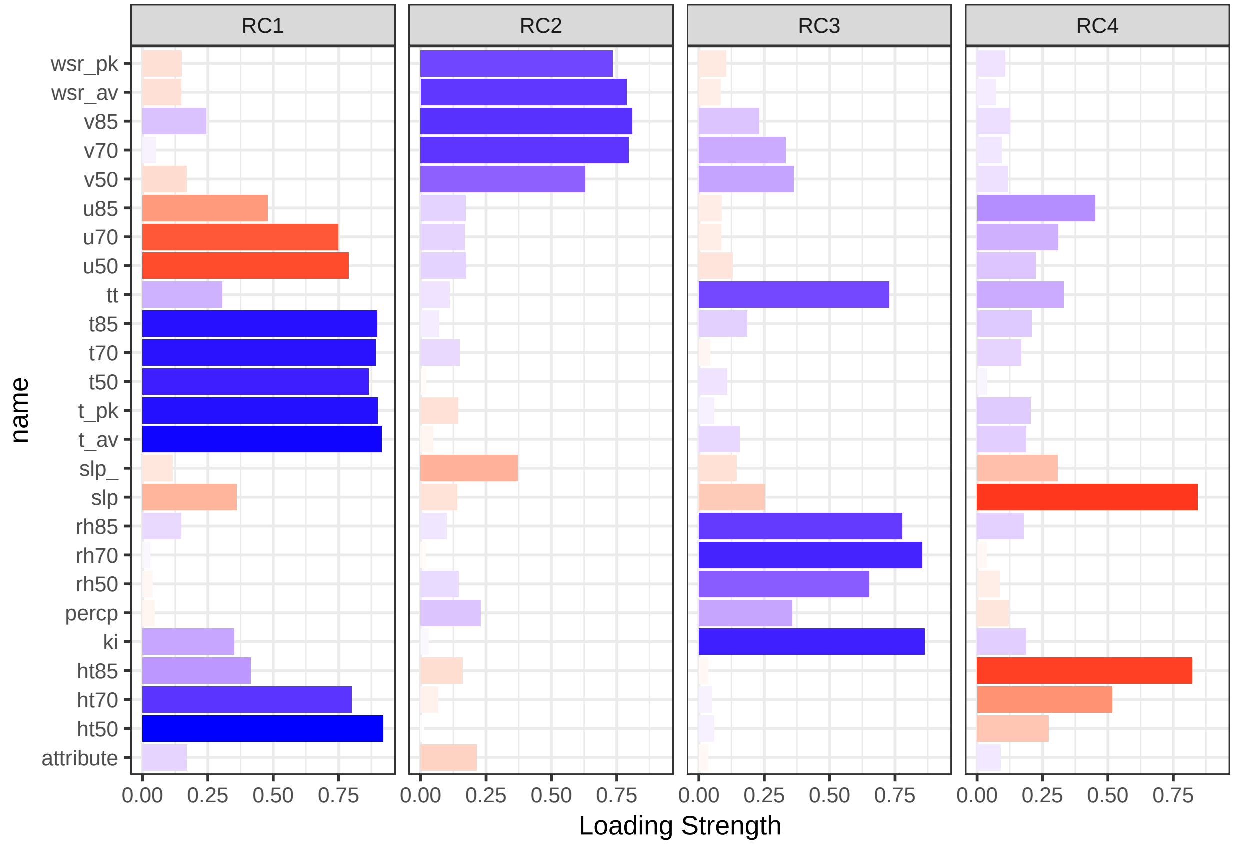
```
## wsr_pk    -0.14967717 -0.10387121  0.73465415  0.10758513    wsr_pk
## wsr_av    -0.14760352 -0.08235590  0.78690179  0.07099203    wsr_av
## t_pk       0.89873480  0.05749781 -0.14365053  0.20524199      t_pk
## t_av       0.91521395  0.15605111 -0.04778759  0.18760338      t_av
## t85        0.89748700  0.18522342  0.07068584  0.20957385       t85
## rh85       0.14727550  0.77654070  0.09927568  0.17897850      rh85
## u85       -0.47986472 -0.08777645  0.17183571  0.45106496       u85
## v85        0.24324192  0.22963282  0.80918884  0.12694184       v85
## ht85       0.41322167 -0.03563537 -0.16055255 -0.82226494      ht85
## t70        0.89263258 -0.04339647  0.14924922  0.16792791       t70
## rh70       0.02939269  0.85284641 -0.01944640 -0.03621383      rh70
## u70       -0.74825724 -0.08581162  0.16885749  0.31108401       u70
## v70        0.05014715  0.33210046  0.79574737  0.09486665       v70
## ht70       0.79926328  0.04918376 -0.06750865 -0.51573659      ht70
## t50        0.86470728  0.10811177 -0.02018109  0.03811741       t50
## rh50      -0.03840935  0.65087708  0.14602291 -0.08573260      rh50
## u50       -0.78853347 -0.12959144  0.17293447  0.22432528       u50
## v50       -0.17002674  0.36179751  0.62947079  0.11784381       v50
## ht50       0.92095887  0.05778277 -0.01225582 -0.27445050      ht50
## ki         0.35030815  0.86283258  0.03014040  0.18845557        ki
## tt         0.30519697  0.72792960  0.11065528  0.33171606        tt
## slp       -0.36097027 -0.25217505 -0.13912173 -0.84337065       slp
## slp_      -0.11624278 -0.14467341 -0.37167785 -0.30768551      slp_
## percp     -0.04604161  0.35575916  0.22953499 -0.12130184     percp
## attribute  0.16924212 -0.03497646 -0.21479997  0.09037418 attribute
```

```r
library(corrplot)
C<-cor(oneav)
corrplot(C, method = "circle", order = "FPC")
```

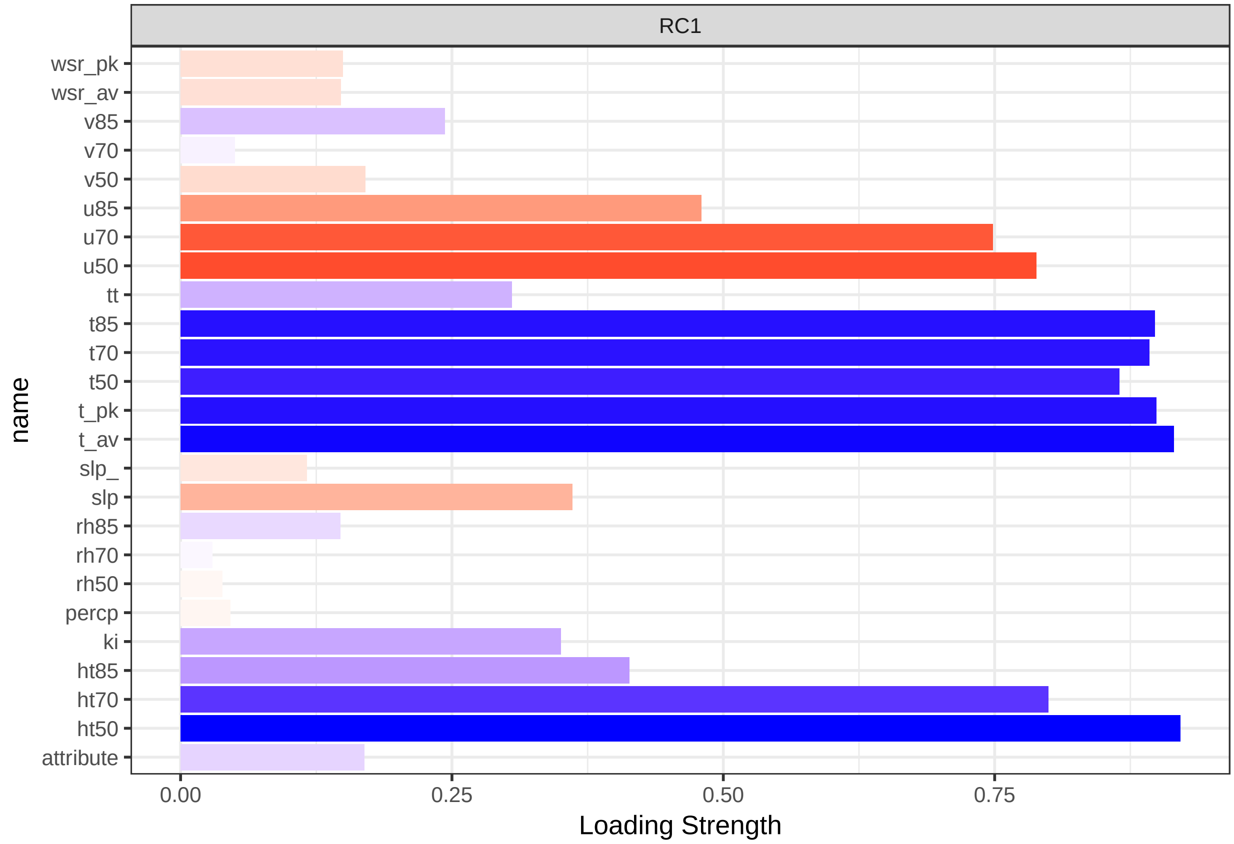


#Visualization of the LOADINGS

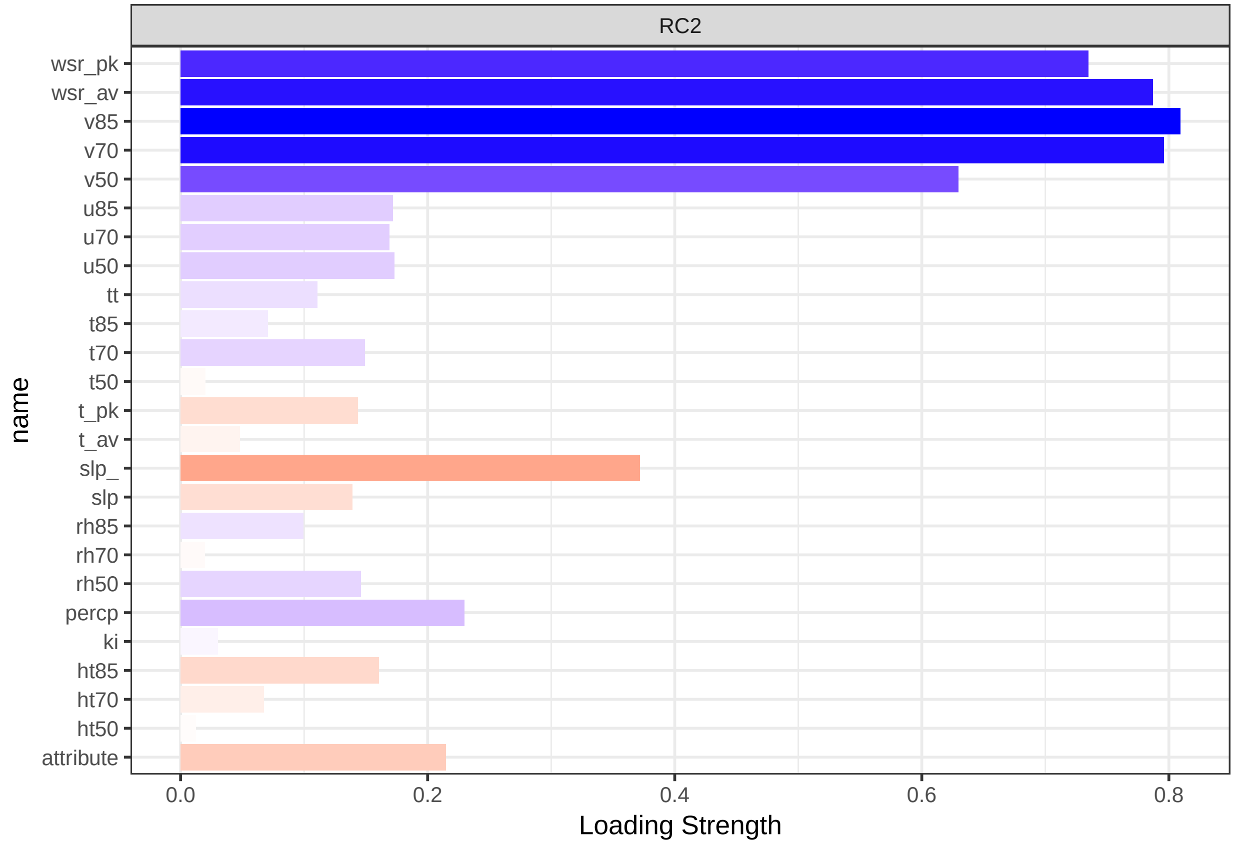
```
#visualization of all loadings
loadings.m <- melt(fitloading, id="name",
                   measure=c("RC1","RC2","RC3","RC4"),
                   variable.name="Factor", value.name="Loading")
ggplot(loadings.m, aes(name, abs(Loading), fill=Loading)) +
  facet_wrap(~ Factor, nrow=1) + #place the factors in separate facets
  geom_bar(stat="identity") + #make the bars
  coord_flip() + #flip the axes so the test names can be horizontal
  #define the fill color gradient: blue=positive, red=negative
  scale_fill_gradient2(name = "Loading",
                       high = "blue", mid = "white", low = "red",
                       midpoint=0, guide=F) +
  ylab("Loading Strength") + #improve y-axis label
  theme_bw(base_size=10) #use a black-andOwhite theme with set font size
```



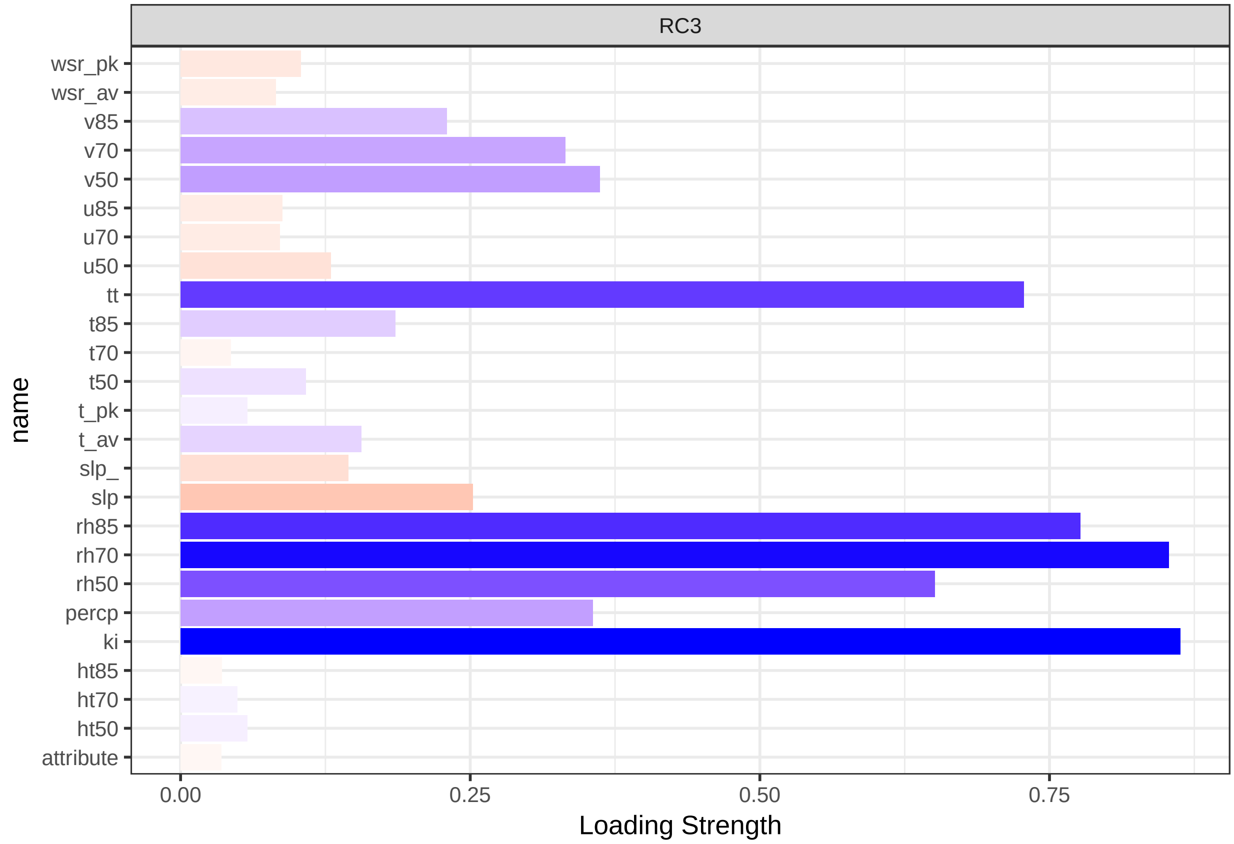

```
#################VISUALIZATION OF THE LOADINGS############################
#Visualization of loading 1

loadings.1 <- melt(fitloading, id="name",
                   measure=c("RC1"),
                   variable.name="Factor", value.name="Loading")
ggplot(loadings.1, aes(name, abs(Loading), fill=Loading)) +
  facet_wrap(~ Factor, nrow=1) + #place the factors in separate facets
  geom_bar(stat="identity") + #make the bars
  coord_flip() + #flip the axes so the test names can be horizontal
  #define the fill color gradient: blue=positive, red=negative
  scale_fill_gradient2(name = "Loading",
                       high = "blue", mid = "white", low = "red",
                       midpoint=0, guide=F) +
  ylab("Loading Strength") + #improve y-axis label
  theme_bw(base_size=10) #use a black-andOwhite theme with set font size
```



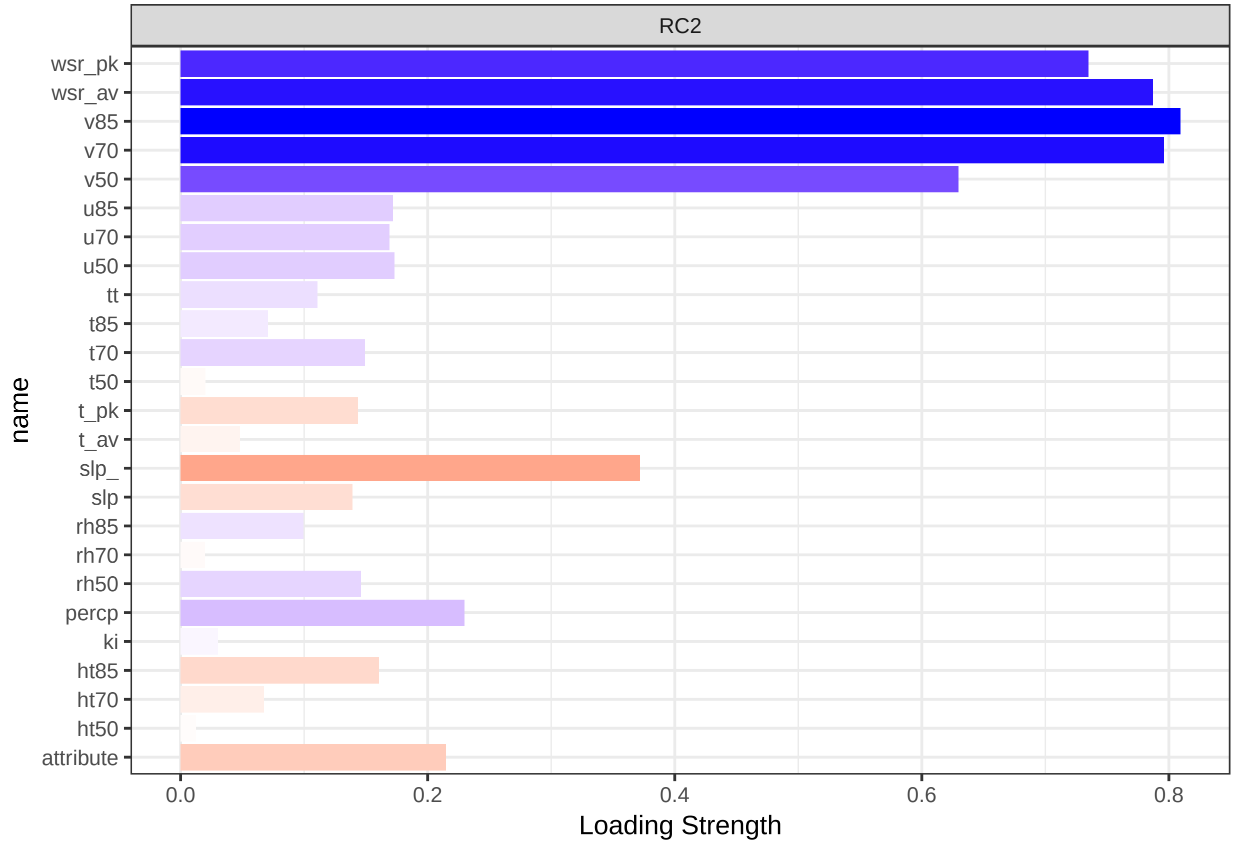

```r
#Visualization of loading 2

loadings.2 <- melt(fitloading, id="name",
                   measure=c("RC2"),
                   variable.name="Factor", value.name="Loading")
ggplot(loadings.2, aes(name, abs(Loading), fill=Loading)) +
  facet_wrap(~ Factor, nrow=1) + #place the factors in separate facets
  geom_bar(stat="identity") + #make the bars
  coord_flip() + #flip the axes so the test names can be horizontal
  #define the fill color gradient: blue=positive, red=negative
  scale_fill_gradient2(name = "Loading",
                       high = "blue", mid = "white", low = "red",
                       midpoint=0, guide=F) +
  ylab("Loading Strength") + #improve y-axis label
  theme_bw(base_size=10) #use a black-andOwhite theme with set font size
```




```
#Visualization of Loading 3
loadings.3 <- melt(fitloading, id="name",
                   measure=c("RC3"),
                   variable.name="Factor", value.name="Loading")
ggplot(loadings.3, aes(name, abs(Loading), fill=Loading)) +
  facet_wrap(~ Factor, nrow=1) + #place the factors in separate facets
  geom_bar(stat="identity") + #make the bars
  coord_flip() + #flip the axes so the test names can be horizontal
  #define the fill color gradient: blue=positive, red=negative
  scale_fill_gradient2(name = "Loading",
                       high = "blue", mid = "white", low = "red",
                       midpoint=0, guide=F) +
  ylab("Loading Strength") + #improve y-axis label
  theme_bw(base_size=10) #use a black-andOwhite theme with set font size
```




```r
#Visualization of Loading 4
loadings.4 <- melt(fitloading, id="name",
                   measure=c("RC4"),
                   variable.name="Factor", value.name="Loading")
ggplot(loadings.2, aes(name, abs(Loading), fill=Loading)) +
  facet_wrap(~ Factor, nrow=1) + #place the factors in separate facets
  geom_bar(stat="identity") + #make the bars
  coord_flip() + #flip the axes so the test names can be horizontal
  #define the fill color gradient: blue=positive, red=negative
  scale_fill_gradient2(name = "Loading",
                       high = "blue", mid = "white", low = "red",
                       midpoint=0, guide=F) +
  ylab("Loading Strength") + #improve y-axis label
  theme_bw(base_size=10) #use a black-andOwhite theme with set font size
```




```r
names(fit)
```

```
##  [1] "values"       "rotation"     "n.obs"        "communality"
##  [5] "loadings"     "fit"          "fit.off"      "fn"
##  [9] "Call"         "uniquenesses" "complexity"   "chi"
## [13] "EPVAL"        "R2"           "objective"    "residual"
## [17] "rms"          "factors"      "dof"          "null.dof"
## [21] "null.model"   "criteria"     "STATISTIC"    "PVAL"
## [25] "weights"      "r.scores"     "rot.mat"      "Vaccounted"
## [29] "Structure"    "scores"
```

```r
fit$loadings
```

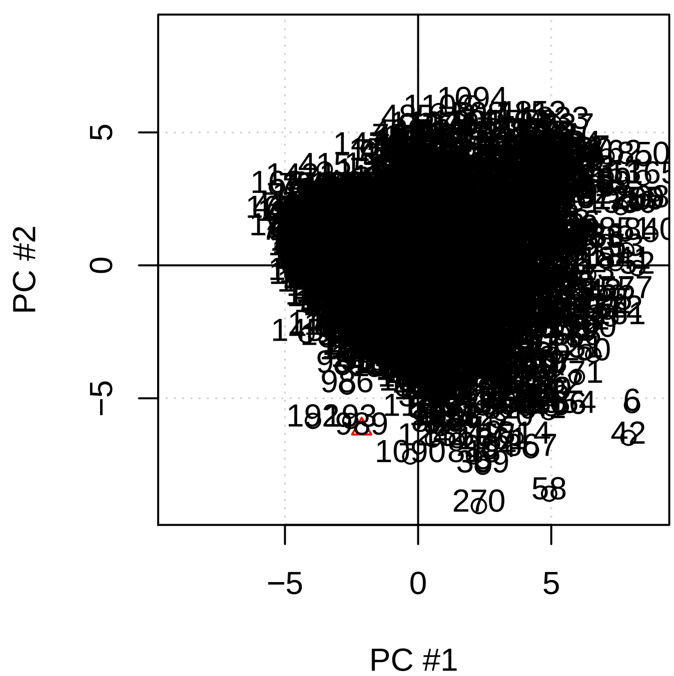
```
##
## Loadings:
##             RC1    RC3    RC2    RC4
## wsr_pk     -0.150 -0.104  0.735  0.108
## wsr_av     -0.148          0.787
## t_pk        0.899         -0.144  0.205
## t_av        0.915  0.156          0.188
## t85         0.897  0.185          0.210
## rh85        0.147  0.777          0.179
## u85        -0.480          0.172  0.451
## v85         0.243  0.230  0.809  0.127
## ht85        0.413         -0.161 -0.822
## t70         0.893          0.149  0.168
## rh70               0.853
```

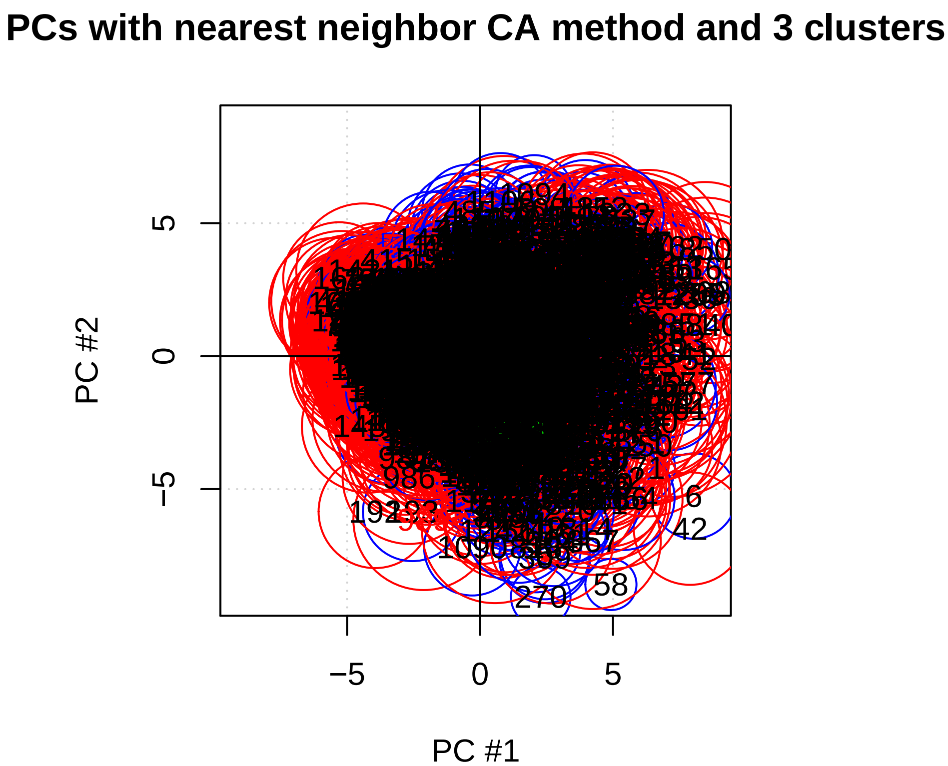
```
## u70       -0.748         0.169  0.311
## v70               0.332  0.796
## ht70       0.799               -0.516
## t50        0.865  0.108
## rh50              0.651  0.146
## u50       -0.789 -0.130  0.173  0.224
## v50       -0.170  0.362  0.629  0.118
## ht50       0.921               -0.274
## ki         0.350  0.863         0.188
## tt         0.305  0.728  0.111  0.332
## slp       -0.361 -0.252 -0.139 -0.843
## slp_      -0.116 -0.145 -0.372 -0.308
## percp             0.356  0.230 -0.121
## attribute  0.169        -0.215
##
##                   RC1   RC3   RC2   RC4
## SS loadings    7.614 3.667 3.313 2.590
## Proportion Var 0.305 0.147 0.133 0.104
## Cumulative Var 0.305 0.451 0.584 0.687
```

```r
#code taken from http://www.chrisbilder.com/multivariate/Section2/goblet_CA.r
PCA.CA.plot<-function(data.set, cluster.results, numb.clust, plot.title, cor.use = TRUE, inches = 0.5) {

  clusters<-cutree(tree = cluster.results, k = numb.clust)

  #PC scores
  pca.cor<-princomp(x = data.set, cor = cor.use, scores = FALSE)
  pca.cor$scale<-apply(X = data.set, MARGIN = 2, FUN = sd)
  score.cor<-predict(pca.cor, newdata = data.set)

  #Scatter plot of first two PCs

  par(pty = "s")
  common.limits<-c(min(score.cor[,1:2]), max(score.cor[,1:2]))
  plot(x = score.cor[,1], y = score.cor[,2], xlab = "PC #1", ylab = "PC #2",
       main = paste("PCs with", plot.title, "and", numb.clust, "clusters"),
       xlim = common.limits, ylim = common.limits, panel.first = grid(col = "lightgray", lty = "dotted")
       col = clusters, pch = clusters)
  abline(h = 0)
  abline(v = 0)
  text(x = score.cor[,1], y = score.cor[,2]+0.2)

  #Bubble plot of first three PCs

  par(pty = "s")
  PC3.positive<-score.cor[,3] - min(score.cor[,3])  #Bubble needs to contain all values > 0
  col.symbol<-ifelse(test = score.cor[,3]>0, yes = "red", no = "blue")
  symbols(x = score.cor[,1], y = score.cor[,2], circles = PC3.positive,
          xlab = "PC #1", ylab = "PC #2", main = paste("PCs with", plot.title, "and", numb.clust, "clus
          xlim = common.limits, ylim = common.limits, panel.first = grid(col = "lightgray", lty = "dott
          fg = col.symbol)
  text(x = score.cor[,1], y = score.cor[,2], col = clusters)
  abline(h = 0)
  abline(v = 0)
```

```
  #3D plot - Note: I used common limits for all three dimensions here because the distance between poin
  plot3d(x = score.cor[,1], y = score.cor[,2], z = score.cor[,3], xlab = "PC #1", ylab = "PC #2",
         zlab = "PC #3", type = "h", xlim = common.limits, ylim = common.limits, zlim = common.limits)
  plot3d(x = score.cor[,1], y = score.cor[,2], z = score.cor[,3], add = TRUE, col = clusters, size = 6)
  persp3d(x = common.limits, y = common.limits, z = matrix(data = c(0,0,0,0), nrow = 2, ncol = 2),
          add = TRUE, col = "green") #Put a plane on the plot
  grid3d(side = c("x", "y", "z"), col = "lightgray")

  invisible()
}

Z<-scale(oneav)
dist.mat<-dist(x = Z, method = "euclidean")
clust.nn<-hclust(d = dist.mat, method = "single")
PCA.CA.plot(data.set = oneav, cluster.results = clust.nn, numb.clust = 3,
            plot.title = "nearest neighbor CA method")
```

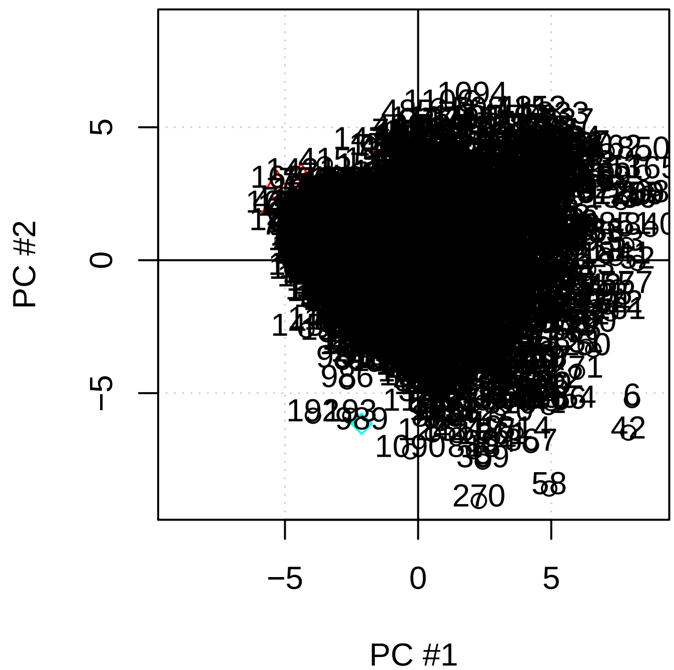
PCs with nearest neighbor CA method and 3 clusters

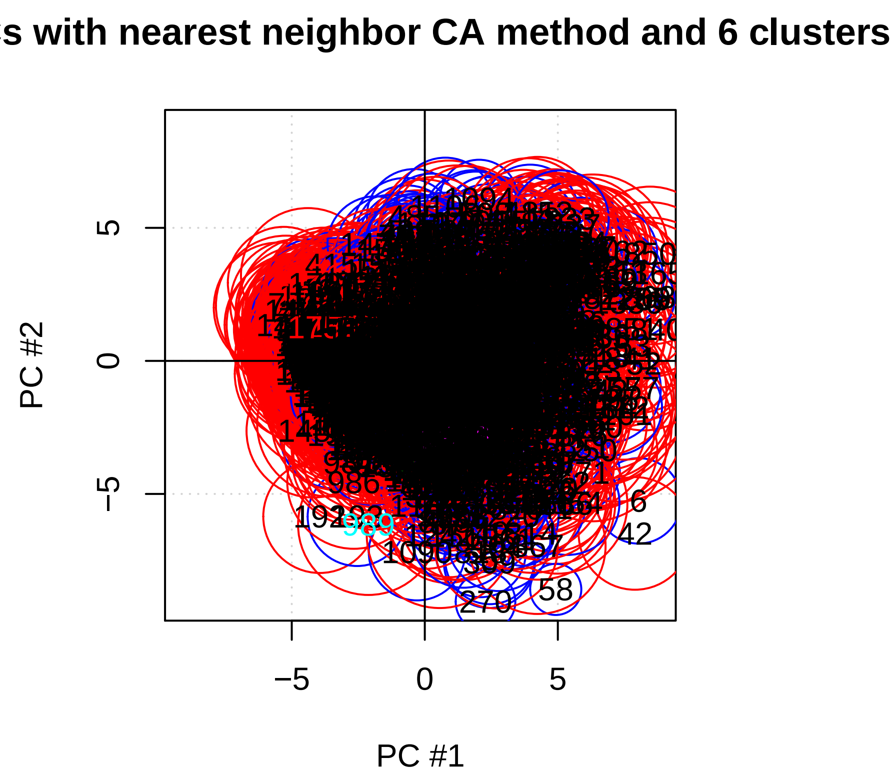





```r
PCA.CA.plot(data.set = oneav, cluster.results = clust.nn, numb.clust = 6,
            plot.title = "nearest neighbor CA method")
```

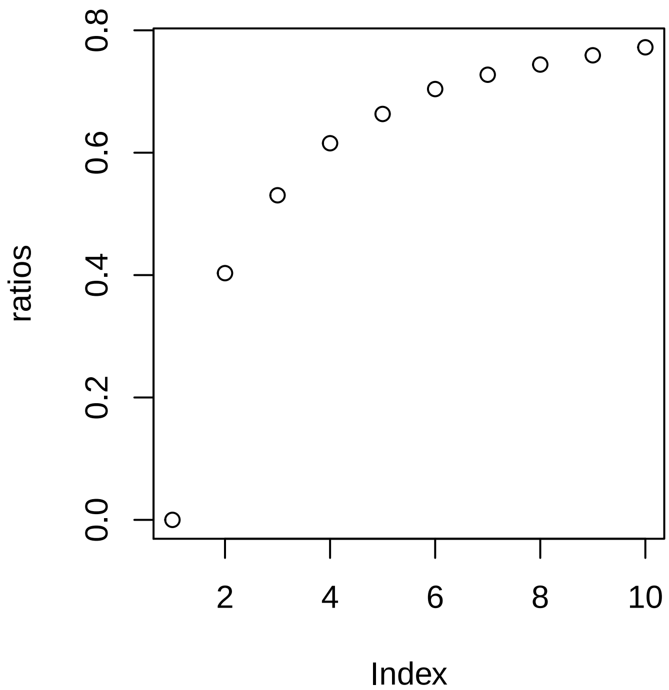
PCs with nearest neighbor CA method and 6 clusters



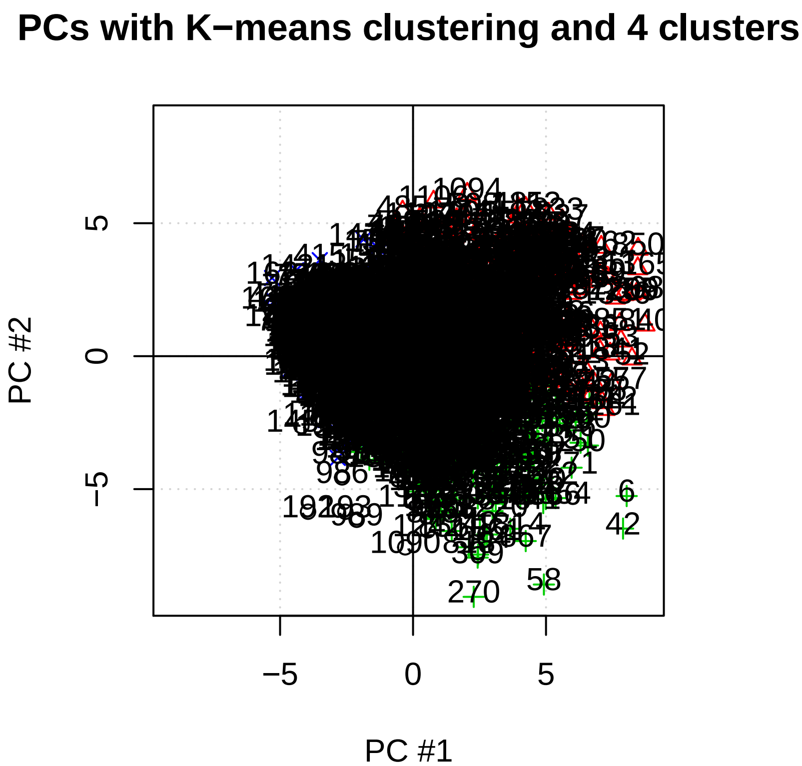


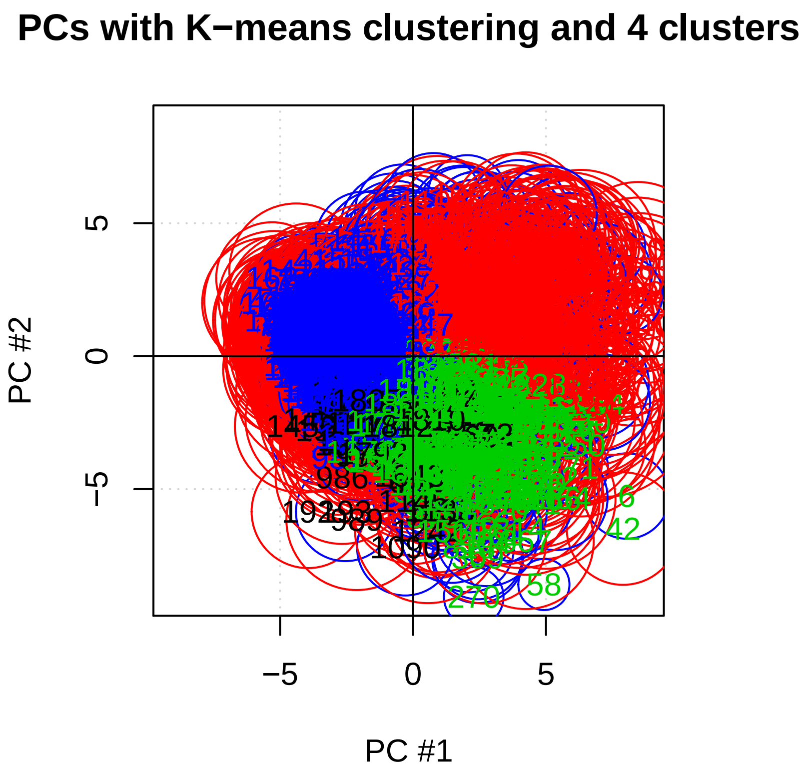

```
##################################### Principal COmponent K-Means Clustering############
PCA.CA.plot2<-function(data.set, cluster.results, plot.title, cor.use = TRUE, inches = 0.5) {

  numb.clust<-length(cluster.results$size)

  #PC scores
  pca.cor<-princomp(x = data.set, cor = cor.use, scores = FALSE)
  pca.cor$scale<-apply(X = data.set, MARGIN = 2, FUN = sd)
  score.cor<-predict(pca.cor, newdata = data.set)

  #Scatter plot of first two PCs


  common.limits<-c(min(score.cor[,1:2]), max(score.cor[,1:2]))
  plot(x = score.cor[,1], y = score.cor[,2], xlab = "PC #1", ylab = "PC #2",
       main = paste("PCs with", plot.title, "and", numb.clust, "clusters"),
       xlim = common.limits, ylim = common.limits, panel.first = grid(col = "lightgray", lty = "dotted")
       col = cluster.results$cluster, pch = cluster.results$cluster)
  abline(h = 0)
  abline(v = 0)
  text(x = score.cor[,1], y = score.cor[,2]+0.2)

  #Bubble plot of first three PCs


  PC3.positive<-score.cor[,3] - min(score.cor[,3])  #Bubble needs to contain all values > 0
```

```r
  col.symbol<-ifelse(test = score.cor[,3]>0, yes = "red", no = "blue")
  symbols(x = score.cor[,1], y = score.cor[,2], circles = PC3.positive,
          xlab = "PC #1", ylab = "PC #2", main = paste("PCs with", plot.title, "and", numb.clust, "clus
          xlim = common.limits, ylim = common.limits, panel.first = grid(col = "lightgray", lty = "dott
          fg = col.symbol)
  text(x = score.cor[,1], y = score.cor[,2], col = cluster.results$cluster)
  abline(h = 0)
  abline(v = 0)

  #3D plot - Note: I used common limits for all three dimensions here because the distance between poin
  plot3d(x = score.cor[,1], y = score.cor[,2], z = score.cor[,3], xlab = "PC #1", ylab = "PC #2",
         zlab = "PC #3", type = "h", xlim = common.limits, ylim = common.limits, zlim = common.limits)
  plot3d(x = score.cor[,1], y = score.cor[,2], z = score.cor[,3], add = TRUE, col = cluster.results$clu
  persp3d(x = common.limits, y = common.limits, z = matrix(data = c(0,0,0,0), nrow = 2, ncol = 2),
          add = TRUE, col = "green") #Put a plane on the plot
  grid3d(side = c("x", "y", "z"), col = "lightgray")

  invisible()
}
# determine the number of clustering
oneav.c <- kmeans(oneav, 3, nstart=10)
ratios <- vector()
for (k in 1:10) {
  oneav.c <- kmeans(oneav, k, nstart=10)
  ratios[k] <- oneav.c$betweenss / oneav.c$totss
}
plot(ratios)
```




```
oneav.c<-kmeans(x = Z, centers = 4)
PCA.CA.plot2(data.set = oneav, cluster.results = oneav.c,
             plot.title = "K-means clustering")
```


PCs with K−means clustering and 4 clusters
PC #2
PC #1
5
0
−5
−5
0
5


PCs with K−means clustering and 4 clusters
PC #2
PC #1
5
0
−5
−5
0
5