

HW4_PCA_Yuefei_Chen

Yuefei Chen

2024-03-03

At first, load Dataset and Preprocess the house rent data

Question 1 Decide how many Principal Components (PCs) you want to keep and why

ANS: In this problem, we decide to choose two PCs. In the following analysis, we observe the scree diagram and it shows before PC2 there are significantly decrease in the line. It means these two components variances are more significant than other values. In addition, in the eigenvalue table, the eigenvalue of PC1 is larger than 1 and PC2 is close to 1. That is also a reason to select these two PCs.

Question 2 Explain the variate representation each PCs

ANS: According to the table of `summary(rent_pca)`, we can find that PC1 has positive relation with every variable. In these variables, the variables area and bathrooms have significant positive with PC1. In addition, PC2 has positive relation with area, rooms, 'bathroom, parking spaces, and fire insurance. These relations are not significant. However, PC2 has significant negative with variables hoa and property tax.

Question 3 Perform some visualization using PCs.

ANS: Click to Visualization

```
library(readr)
house_data <- read_csv("Dataset/Rent_House_random_200.csv")

## Rows: 200 Columns: 11
## -- Column specification -----
## Delimiter: ","
## chr (3): floor, animal, furniture
## dbl (8): area, rooms, bathroom, parking spaces, hoa, rent amount, property t...
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.

house_data <- house_data[, c(1:4, 6:11)]
str(house_data)
```

```
## tibble [200 x 10] (S3: tbl_df/tbl/data.frame)
## $ area      : num [1:200] 120 45 50 35 204 177 15 70 180 180 ...
## $ rooms     : num [1:200] 3 1 2 1 4 3 1 2 3 4 ...
## $ bathroom  : num [1:200] 4 1 1 1 4 3 1 2 3 4 ...
## $ parking spaces: num [1:200] 3 1 1 0 2 4 0 1 2 2 ...
## $ animal    : chr [1:200] "accept" "not accept" "accept" "accept" ...
## $ furniture  : chr [1:200] "not furnished" "furnished" "not furnished" "not furnished" ...
## $ hoa       : num [1:200] 1350 3000 226 260 0 2700 0 1800 700 2600 ...
## $ rent amount : num [1:200] 5600 5520 750 1400 3440 6900 1200 4200 2700 2000 ...
## $ property tax : num [1:200] 560 0 0 0 100 509 0 250 175 584 ...
## $ fire insurance: num [1:200] 71 70 10 18 62 89 16 55 40 26 ...
```

```
# Using prcomp to compute the principal components (eigenvalues and eigenvectors). With scale=TRUE, var
rent_pca <- prcomp(house_data[, -c(5,6,8)], scale=TRUE)
rent_pca
```

```
## Standard deviations (1, ..., p=7):
## [1] 2.1011414 0.9226665 0.7787804 0.6504630 0.5857012 0.4749530 0.3683260
##
## Rotation (n x k) = (7 x 7):
##          PC1      PC2      PC3      PC4      PC5
## area      0.4336965 0.17091900 0.11991336 -0.14262889 -0.35241854
## rooms     0.3704037 0.38742007 0.06788605 0.73084114 0.02399273
## bathroom  0.4227960 0.15626819 0.06607019 0.09708057 0.28600555
## parking spaces 0.3939576 0.08267643 0.27820446 -0.52057449 0.56700711
## hoa       0.3144122 -0.55087277 -0.64250123 0.17481916 0.31296716
## property tax 0.2933799 -0.68199336 0.54402406 0.12426281 -0.30542261
## fire insurance 0.3947438 0.14443506 -0.43643932 -0.34511274 -0.52980928
##          PC6      PC7
## area      -0.04502695 0.78850847
## rooms     0.39473933 -0.13256958
## bathroom  -0.81830269 -0.17780729
## parking spaces 0.39883150 -0.09488344
## hoa       0.08754800 0.22068420
## property tax 0.01872523 -0.20922784
## fire insurance 0.07383965 -0.47705723
```

```
summary(rent_pca)
```

```
## Importance of components:
##          PC1      PC2      PC3      PC4      PC5      PC6      PC7
## Standard deviation      2.1011 0.9227 0.77878 0.65046 0.58570 0.47495 0.36833
## Proportion of Variance 0.6307 0.1216 0.08664 0.06044 0.04901 0.03223 0.01938
## Cumulative Proportion 0.6307 0.7523 0.83894 0.89939 0.94839 0.98062 1.00000
```

```
(eigen_rent <- rent_pca$sdev^2)
```

```
## [1] 4.4147952 0.8513135 0.6064989 0.4231021 0.3430459 0.2255804 0.1356640
```

```
names(eigen_rent) <- paste("PC", 1:7, sep="")
eigen_rent
```

```
##      PC1      PC2      PC3      PC4      PC5      PC6      PC7
## 4.4147952 0.8513135 0.6064989 0.4231021 0.3430459 0.2255804 0.1356640
```

```
sumlambdas <- sum(eigen_rent)
sumlambdas
```

```
## [1] 7
```

```
propvar <- eigen_rent/sumlambdas
propvar
```

```
##      PC1      PC2      PC3      PC4      PC5      PC6      PC7
## 0.63068502 0.12161621 0.08664270 0.06044316 0.04900656 0.03222577 0.01938057
```

```
cumvar_rent <- cumsum(propvar)
cumvar_rent
```

```
##      PC1      PC2      PC3      PC4      PC5      PC6      PC7
## 0.6306850 0.7523012 0.8389439 0.8993871 0.9483937 0.9806194 1.0000000
```

```
matlambdas <- rbind(eigen_rent,propvar,cumvar_rent)
rownames(matlambdas) <- c("Eigenvalues","Prop. variance","Cum. prop. variance")
round(matlambdas,4)
```

```
##      PC1      PC2      PC3      PC4      PC5      PC6      PC7
## Eigenvalues      4.4148 0.8513 0.6065 0.4231 0.3430 0.2256 0.1357
## Prop. variance    0.6307 0.1216 0.0866 0.0604 0.0490 0.0322 0.0194
## Cum. prop. variance 0.6307 0.7523 0.8389 0.8994 0.9484 0.9806 1.0000
```

```
summary(rent_pca)
```

```
## Importance of components:
```

```
##      PC1      PC2      PC3      PC4      PC5      PC6      PC7
## Standard deviation 2.1011 0.9227 0.77878 0.65046 0.58570 0.47495 0.36833
## Proportion of Variance 0.6307 0.1216 0.08664 0.06044 0.04901 0.03223 0.01938
## Cumulative Proportion 0.6307 0.7523 0.83894 0.89939 0.94839 0.98062 1.00000
```

```
rent_pca$rotation
```

```
##      PC1      PC2      PC3      PC4      PC5
## area      0.4336965 0.17091900 0.11991336 -0.14262889 -0.35241854
## rooms      0.3704037 0.38742007 0.06788605 0.73084114 0.02399273
## bathroom    0.4227960 0.15626819 0.06607019 0.09708057 0.28600555
## parking spaces 0.3939576 0.08267643 0.27820446 -0.52057449 0.56700711
## hoa      0.3144122 -0.55087277 -0.64250123 0.17481916 0.31296716
## property tax 0.2933799 -0.68199336 0.54402406 0.12426281 -0.30542261
## fire insurance 0.3947438 0.14443506 -0.43643932 -0.34511274 -0.52980928
##      PC6      PC7
## area     -0.04502695 0.78850847
## rooms      0.39473933 -0.13256958
```

```
## bathroom      -0.81830269 -0.17780729
## parking spaces 0.39883150 -0.09488344
## hoa           0.08754800  0.22068420
## property tax   0.01872523 -0.20922784
## fire insurance 0.07383965 -0.47705723
```

```
print(rent_pca)
```

```
## Standard deviations (1, ..., p=7):
## [1] 2.1011414 0.9226665 0.7787804 0.6504630 0.5857012 0.4749530 0.3683260
##
## Rotation (n x k) = (7 x 7):
##
##          PC1      PC2      PC3      PC4      PC5
## area      0.4336965  0.17091900  0.11991336 -0.14262889 -0.35241854
## rooms     0.3704037  0.38742007  0.06788605  0.73084114  0.02399273
## bathroom  0.4227960  0.15626819  0.06607019  0.09708057  0.28600555
## parking spaces 0.3939576  0.08267643  0.27820446 -0.52057449  0.56700711
## hoa       0.3144122 -0.55087277 -0.64250123  0.17481916  0.31296716
## property tax 0.2933799 -0.68199336  0.54402406  0.12426281 -0.30542261
## fire insurance 0.3947438  0.14443506 -0.43643932 -0.34511274 -0.52980928
##
##          PC6      PC7
## area     -0.04502695  0.78850847
## rooms     0.39473933 -0.13256958
## bathroom -0.81830269 -0.17780729
## parking spaces 0.39883150 -0.09488344
## hoa       0.08754800  0.22068420
## property tax 0.01872523 -0.20922784
## fire insurance 0.07383965 -0.47705723
```

```
house_data$rent <- ifelse(house_data$`rent amount` > 3891, "high", "low")
# Identifying the scores by their rent status
rent_amount_pca <- cbind(data.frame(house_data$`rent`), rent_pca$x)
rent_amount_pca
```

```
##      house_data.rent      PC1      PC2      PC3      PC4
## 1      high  1.56783825  0.216106460  0.1006524694 -0.121661131
## 2      high -0.57639521 -1.415621716 -1.8450487308 -0.568054942
## 3      low  -1.50346548  0.047393945  0.3146626557  0.075959005
## 4      low  -2.07514390 -0.353409399 -0.0487786567 -0.213547579
## 5      low   1.34530122  1.543191001  0.6261747058  0.566419730
## 6      high  2.24975917 -0.304496824 -0.6359827011 -0.569522889
## 7      low  -2.23846999 -0.266490242  0.0926178033 -0.212726044
## 8      high -0.22912144 -0.584847508 -0.7695133831  0.056490473
## 9      low   0.64873235  0.609778758  0.3561100132  0.182018436
## 10     low   1.79476843 -0.172057539 -0.2280405746  1.316321010
## 11     low  -0.95873902 -0.028834196  0.1474446455  0.170387563
## 12     high  2.13362330 -0.340120927  0.3471215144 -0.028438097
## 13     low  -1.01660046  0.184564734  0.2334791298  0.244747393
## 14     low  -2.07539668 -0.503254179 -0.1877055263 -0.142237669
## 15     low  -1.35466295  0.014694313  0.2082497129  0.034576983
## 16     low  -0.76938889  0.177445816 -0.0117026192  1.164270693
## 17     high -0.82924339 -0.283533687 -0.3362267145 -0.808440700
## 18     low  -1.67879982 -0.732653505 -0.4699507755 -0.195224545
```

## 19	high	0.63656545	0.254965890	-0.3726995379	0.470457449
## 20	low	-2.03744269	-0.436072675	-0.0208544152	-0.173039886
## 21	low	-1.54280272	0.162255561	0.4101846230	0.020917191
## 22	high	0.16777452	0.541686279	0.0862675807	0.016597064
## 23	high	3.19501820	1.625800103	1.2208364392	-0.437836947
## 24	high	3.50284784	2.828627579	-0.4270439877	3.021320433
## 25	low	-0.32699280	0.668985737	0.3654347279	0.148866006
## 26	low	-1.89708178	-0.561270151	-0.2264566344	-0.146433688
## 27	low	-2.05561487	-0.358852444	-0.0833815781	-0.235174184
## 28	low	-1.89149015	-0.180148663	0.3430894004	-0.588317122
## 29	low	-0.87692112	-0.111563708	0.1528911503	0.165100917
## 30	high	1.85262375	1.096409200	0.3129228017	-0.494026791
## 31	low	-1.88522590	-0.535453098	-0.3545206600	-0.203286079
## 32	high	1.13964853	-0.038181296	-0.2171061084	0.849371034
## 33	high	1.30826933	0.088697159	0.2887956702	-0.145902126
## 34	low	-1.22973070	0.051707352	0.1518058641	-0.023532342
## 35	low	2.24230187	0.498753274	0.2538800688	0.748496378
## 36	high	1.94733013	-0.376827121	0.4992531656	0.436069290
## 37	high	8.72831120	-4.001864821	-1.7044282832	0.278709593
## 38	low	-0.61880179	0.702893459	0.5867092275	0.613271626
## 39	low	-2.02535867	-0.502475425	-0.2403781125	-0.169854715
## 40	low	0.07631034	0.738460865	0.2501213215	1.207803582
## 41	low	-2.08583746	-0.234423820	0.1273052975	-0.264569312
## 42	high	2.94958392	0.427892318	-0.2277217426	-0.057963432
## 43	low	0.68623055	0.197753792	0.0728335361	0.303568344
## 44	low	0.27919326	0.633781791	0.2892257520	0.194552423
## 45	low	1.10879850	0.272393159	-0.1253531306	1.284610499
## 46	high	2.05071676	1.472483035	1.1316154236	-1.112095378
## 47	high	4.21959089	0.100950841	-0.9469743675	-0.648501558
## 48	low	-1.27087429	0.301136038	0.0872555875	0.989808541
## 49	high	4.58113442	-0.081008408	-3.0702617795	-1.484704799
## 50	low	-2.02223866	-0.307741537	-0.0026440759	-0.247242514
## 51	low	-0.42267359	0.242911565	0.0287822969	0.727401222
## 52	low	-0.90365748	0.104475522	0.1721109085	0.035788515
## 53	low	-0.58003387	0.329728014	0.2223298923	0.705890676
## 54	high	2.90035484	-1.570597144	0.1787300097	0.048382923
## 55	low	6.70167586	-7.240368801	5.7122500453	1.569939729
## 56	low	-1.58788243	-0.152880120	-0.1212449872	0.445355642
## 57	low	-2.08095556	-0.361259848	-0.0448878179	-0.204601143
## 58	low	-2.07987411	-0.204708260	0.1011651456	-0.278814209
## 59	low	-1.23803192	-0.239285062	0.1066257807	0.127084933
## 60	low	-1.27346638	0.227541583	0.2940400969	-0.141133413
## 61	low	-2.06371912	-0.398381045	0.0192251237	-0.166788952
## 62	low	0.37233762	-0.654972412	0.0250325045	-0.125326341
## 63	high	0.16309312	0.463109560	-0.5870658673	0.724792365
## 64	low	-1.99481328	-0.230902340	0.0164564910	-0.346782693
## 65	high	1.30865679	1.223530239	-0.2185786966	-0.673072950
## 66	low	-0.39000716	0.515254147	0.3673203710	0.577268698
## 67	low	0.64357804	0.421824081	0.0884939598	0.294100301
## 68	low	-1.72324081	-0.400853043	0.1446466693	-0.547683814
## 69	low	-2.01219554	-0.452081716	-0.1779667165	-0.222038946
## 70	low	-2.11306394	-0.376864287	0.0432032659	-0.155543996
## 71	low	-1.60270788	-0.107661056	0.0978338074	0.421153939
## 72	low	0.77543632	0.840639563	0.8125250938	-0.213977712

## 73	low	0.52578996	1.201643439	0.5722349371	0.848863034
## 74	low	-2.19000223	-0.275806650	0.1225112186	-0.220800775
## 75	high	0.28960911	0.175849250	-0.0345351070	0.145994843
## 76	low	-1.76754748	-0.173722835	0.0623678331	-0.426690608
## 77	low	-1.21365160	0.162691257	0.3434054469	0.143526752
## 78	low	-1.80275583	-0.328698732	0.2454104797	-0.553728334
## 79	low	-1.75325429	-0.360381071	-0.3307310625	-0.382124982
## 80	low	0.80748929	0.472345300	-0.1040745716	0.796495848
## 81	low	-0.80315118	0.691862669	0.3270645972	0.913433400
## 82	low	-1.13014395	0.104196186	0.3472714236	0.147257619
## 83	low	0.19883740	0.908240616	0.8573177480	-0.282468335
## 84	low	-0.85903456	0.011756540	0.9862872872	-1.751171429
## 85	low	-0.30037935	0.591144964	0.6741041657	0.263472446
## 86	low	-1.47075763	0.098284736	0.2957358096	0.021078241
## 87	high	-0.22483293	0.395265595	-0.2209099434	-1.672353342
## 88	low	-2.28955542	-0.285182186	0.1490992353	-0.168063565
## 89	low	-0.41144395	0.625718703	0.2050961194	0.149367786
## 90	high	1.51846473	0.279585178	-0.4708986879	0.795976786
## 91	low	0.89449869	1.117748417	0.3304967504	0.553363763
## 92	low	-1.53674605	-0.100957919	0.3801463564	-0.210659210
## 93	high	6.19640421	-0.110241455	3.3822304210	-2.805589326
## 94	low	-1.86005657	-0.474529161	-0.2076998800	-0.273596003
## 95	low	0.33212522	0.226532434	-0.1097068019	0.694394101
## 96	low	0.80830540	0.838485361	0.6027470151	0.015415555
## 97	low	1.13265608	-0.765402951	-0.0569401156	-0.641086478
## 98	low	-0.91478591	0.051041051	0.1060233162	0.077531582
## 99	low	-1.03005630	0.069827300	0.2207939449	0.128116383
## 100	low	-0.22835744	-1.993479512	-2.3805418101	0.223876458
## 101	high	4.55935064	-1.051257329	-2.0964459441	1.006330448
## 102	low	-1.78927235	-0.679650433	-0.4539693735	-0.171252851
## 103	low	-1.46833723	-0.425998182	0.1282182677	0.503537374
## 104	low	-1.86780325	-0.411382074	-0.1654655143	-0.233627064
## 105	low	-1.58158590	-0.406008908	0.0230445862	-0.618203957
## 106	high	3.40634597	-0.323061949	-0.4452667611	0.240925817
## 107	high	2.81079495	0.147142954	-1.3679600097	-0.508256718
## 108	low	-1.76194430	0.021059052	0.1047786588	0.378895520
## 109	low	-1.95834325	-0.612496796	-0.2284980060	-0.164177316
## 110	low	-1.47171259	-0.073388736	-0.0185860717	0.350698744
## 111	low	-2.01386835	-0.403640779	-0.1062979106	-0.220561376
## 112	high	5.76211882	-1.137833269	0.3216977116	-0.247423151
## 113	low	-1.71655905	-0.452427504	0.0243278629	-0.545070856
## 114	low	-1.36509520	-0.378424410	-0.0797508969	-0.714234029
## 115	low	-0.65189695	-0.113236592	-0.1763077052	0.514064482
## 116	low	0.13935881	-0.071811510	-0.3566526532	0.629810148
## 117	high	1.93908906	-0.749329254	-1.0016189606	-0.161833200
## 118	low	-0.36387251	0.327039239	-0.0640485327	0.657436626
## 119	high	3.06217521	-0.102970811	-0.8247131126	-0.357370907
## 120	low	0.21859561	0.037872853	-0.1246301221	0.284608250
## 121	low	-1.76243407	0.010817983	0.1048273494	0.416388145
## 122	high	-0.43837393	-0.240555129	-0.6768066510	-1.063240405
## 123	low	-1.02785008	0.128444776	0.0674839263	-0.191208139
## 124	low	-2.09994961	-0.213340349	0.0603113810	-0.286143074
## 125	low	-2.05366442	-0.454152520	0.0653452736	-0.154271356
## 126	low	-2.13691077	-0.322228158	0.1995362903	-0.187430663

## 127	low	-1.79886574	0.064560778	0.2479554409	0.393797437
## 128	low	-0.73346935	0.412906610	-0.0015018796	0.463288506
## 129	low	-0.52651398	0.383270016	0.1459561546	0.668166167
## 130	low	-1.76290055	-0.461315238	0.1287221991	-0.474252553
## 131	low	-2.19205769	-0.288075246	0.1643141751	-0.202823134
## 132	low	-1.60079651	0.154981538	0.2241031268	0.296976666
## 133	low	-0.94453039	0.304808873	0.1088423433	0.623599796
## 134	high	-0.57639521	-1.415621716	-1.8450487308	-0.568054942
## 135	high	4.53213015	-0.402724703	-1.9711126384	-0.192299028
## 136	high	-1.20831130	-0.490355876	-0.3674765904	-0.813156009
## 137	low	-1.99402495	-0.438325527	-0.1996207950	-0.239338848
## 138	low	-1.74622046	-0.258226538	0.1295033127	-0.101829169
## 139	low	-0.65198241	-0.041097747	-0.0720391714	-0.021595749
## 140	high	2.05137269	0.095572333	0.0061182915	0.405188976
## 141	high	0.87380363	0.576758767	0.0688528052	0.713976175
## 142	low	-2.12397932	-0.245487379	0.1501776037	-0.238993450
## 143	low	0.96227605	1.034485851	0.7930384412	0.071989745
## 144	low	-1.01666875	0.081091724	0.2266674509	0.087505628
## 145	high	1.57023222	0.255435229	-0.3052585402	0.214625907
## 146	low	-0.04858875	0.830737518	0.4734066649	-0.001857213
## 147	high	3.68932809	2.143094773	0.7070134257	-0.318375593
## 148	low	-1.91380925	-0.341069088	-0.0135300355	-0.226507296
## 149	low	-0.56676741	0.197178457	0.2227275227	-0.391935961
## 150	low	-2.21839448	-0.257858154	0.1334715678	-0.205397179
## 151	high	-1.28894407	-0.557821845	-0.4672134385	-0.734662855
## 152	high	-0.15718144	0.594554690	0.4669112596	-1.208489344
## 153	low	-1.38189083	0.055106991	0.2532240009	0.011308292
## 154	low	-0.47778490	0.278574381	0.3785521091	-0.509419539
## 155	low	-2.14612555	-0.268589740	0.1643039031	-0.223021844
## 156	high	2.62293011	0.046505699	-1.9347451634	-0.793511580
## 157	low	-0.31225768	0.644030033	0.4338849246	0.216917926
## 158	high	3.76474618	0.110565101	-1.2699536885	0.152972777
## 159	high	0.85651631	0.294530133	-0.9211415805	0.389242523
## 160	high	2.11184808	2.050942284	0.5521557089	1.261666456
## 161	low	-2.15956186	-0.389180550	0.0104111649	-0.157618095
## 162	low	-1.34879407	-0.009258708	0.1083837634	0.015350633
## 163	low	-0.60762535	0.355905635	0.2887305717	0.731005641
## 164	high	2.02586190	0.173151622	-0.8155604445	-0.287006039
## 165	low	-1.05921271	-1.257469439	-0.6819339681	-0.446180451
## 166	low	-1.03323781	-0.198548314	-0.2861963706	-0.009953744
## 167	low	-0.10756112	0.493888032	0.5229805929	0.233120262
## 168	high	3.97320720	1.661466488	1.3303842900	-0.207147700
## 169	high	6.40500435	2.156126384	1.5470106528	-2.330872365
## 170	low	-1.56351179	-0.478888242	-0.0008992945	-0.567448765
## 171	high	4.70912592	1.959217051	0.8330720923	-1.051735715
## 172	low	0.91652794	-0.157807836	0.0695329636	-0.663844743
## 173	high	3.01780491	-3.781663903	0.8357036000	0.868639547
## 174	low	-0.14094557	0.067412311	-0.1700131624	0.659285684
## 175	low	-1.95900629	-0.206996774	0.3126540808	-0.570756911
## 176	high	1.89765204	1.621983337	-0.3189579339	-0.978750094
## 177	low	-2.16356341	-0.298112009	0.0357264063	-0.216893534
## 178	low	-2.04455537	-0.454658134	-0.1167689509	-0.157920212
## 179	low	-0.08406563	0.339842914	-0.0294969343	0.486867554
## 180	low	-1.87051599	-0.556010555	-0.1849082846	-0.150590752

## 181	low	-0.15840553	0.298411747	-0.3890680629	1.023969590
## 182	high	2.57672105	0.835200666	-0.1981408254	0.588203116
## 183	high	1.93043286	1.468098200	1.3121416688	-0.569814917
## 184	low	-0.21811819	-0.062551593	-0.1622152608	0.782733201
## 185	low	-1.66127191	-0.157066582	0.0230435231	-0.487630521
## 186	low	-1.84040471	-0.161456719	0.2866079684	-0.632979600
## 187	high	4.40504874	0.108932928	-1.3397219418	-0.652162530
## 188	low	0.27881638	0.134381161	0.0187600095	-0.380996609
## 189	low	-0.16493909	0.509384595	-0.2625570830	1.671940700
## 190	low	-0.50611546	-0.040305850	0.3322741409	-0.289280669
## 191	high	0.65734764	0.157810165	-0.2833824679	0.559127745
## 192	high	0.96346403	0.745828560	-0.5786646081	0.847574043
## 193	high	3.16873531	0.056330941	-1.2876187644	-0.090831208
## 194	high	1.58293083	0.159448603	-0.1721118343	-0.350725312
## 195	low	-1.43496833	0.162715787	0.3929544973	-0.031172825
## 196	low	-1.41999697	-0.001964711	0.2589000227	0.075919164
## 197	high	5.01064356	-0.837240083	-1.2685998359	-0.107233999
## 198	low	-0.49933014	0.370414441	0.3463092433	0.713629506
## 199	high	-1.33180077	0.285777235	-0.1098022854	0.046297098
## 200	low	-1.21380272	-0.144475181	-0.0122758957	0.022187477
##	PC5	PC6	PC7		
## 1	0.8826685806	-0.37155494	-0.652309477		
## 2	0.2647664146	0.36757198	0.028219570		
## 3	0.2090874646	0.39642134	0.041396776		
## 4	-0.2465915926	-0.19841094	0.041238497		
## 5	0.1268736556	-0.47518491	-0.182963676		
## 6	1.0661267537	0.62606338	-0.110920625		
## 7	-0.2318992105	-0.21326856	-0.125734057		
## 8	0.1919455610	-0.02649857	-0.170601123		
## 9	0.3838699556	-0.17186887	0.239420212		
## 10	1.1613661658	-0.31816341	0.411704385		
## 11	0.4262656529	-0.18383886	0.087079082		
## 12	0.6261856288	-0.36662189	-0.088591731		
## 13	-0.5975235893	-0.48133022	-0.068057089		
## 14	-0.1108141793	-0.17658010	0.014186128		
## 15	0.1255709986	0.41042693	0.105556705		
## 16	0.0106408429	-0.13154428	0.292917274		
## 17	-0.1922592484	-0.43330274	-0.389826301		
## 18	-0.2779252956	-0.12803014	0.129572181		
## 19	-0.1307831555	-0.34796916	-0.232051718		
## 20	-0.2191044191	-0.20007967	0.142622182		
## 21	0.1170964147	0.38458669	-0.039254413		
## 22	0.1308195853	0.49166451	-0.423574239		
## 23	0.1360870018	-0.50326397	-0.087159083		
## 24	-2.0024309975	0.44926000	-0.653456230		
## 25	0.4351206934	0.44997384	-0.630877417		
## 26	-0.1647398582	-0.17741818	0.287627604		
## 27	-0.2815196013	-0.18978103	-0.037672808		
## 28	0.1245282994	0.05300405	0.011093252		
## 29	0.3635219262	-0.17448084	0.064066335		
## 30	-0.8761619265	-0.71570141	-0.571179831		
## 31	-0.2174839032	-0.15533492	0.073410553		
## 32	0.2106339194	0.91449173	-0.097836840		
## 33	0.6530800637	0.19923080	-0.129166151		

## 34	0.0220761334	0.41322082	0.203122550
## 35	1.3711438884	-0.67395345	-0.282508215
## 36	1.6133906820	-1.02080069	0.153092432
## 37	-0.5935284150	-0.39561704	1.567565032
## 38	0.1006836631	0.08545129	0.180486246
## 39	-0.1543627260	-0.16866600	-0.007268863
## 40	0.0295829676	0.48586158	0.107011958
## 41	-0.3672647161	-0.22245958	0.074527502
## 42	0.9230689466	0.31233732	-0.350275555
## 43	0.5637742368	-0.10336173	-0.098328893
## 44	0.5236697654	-0.13869356	-0.564005262
## 45	0.0365471459	-0.04026277	0.295613153
## 46	0.0451082285	0.28839711	1.249763991
## 47	-0.3487199230	0.49152609	-1.077787696
## 48	-0.3266469601	0.46463479	0.056246409
## 49	-0.0565338325	0.31815913	-0.355696246
## 50	-0.3242271864	-0.20702495	0.120685231
## 51	0.3223890632	0.17600324	0.037461623
## 52	0.1937016765	-0.17886050	-0.126102949
## 53	0.2712731924	0.16211897	-0.218563103
## 54	0.0806018599	0.32607158	0.758849158
## 55	-0.7414408847	0.13034774	-1.103420929
## 56	-0.1869680993	0.14648102	0.217575546
## 57	-0.2324739566	-0.19925106	0.053425088
## 58	-0.3742996372	-0.22230437	0.085302417
## 59	0.2218358453	0.43189101	0.228480862
## 60	-0.1859940546	0.40345581	-0.047439743
## 61	-0.2022809065	-0.21046918	0.206448167
## 62	0.7076170094	-0.39211112	-0.285823917
## 63	-0.5955976030	-0.62831523	-0.573206379
## 64	-0.5023016837	-0.19791150	-0.122198137
## 65	-0.9611509766	-0.05590851	-1.029557223
## 66	-0.0057205691	0.13109536	0.077218724
## 67	0.6366932422	-0.14178246	0.291938021
## 68	0.1597990979	0.08696266	0.066090177
## 69	-0.2643086468	-0.17064303	-0.104731833
## 70	-0.1707213827	-0.21542999	0.196668739
## 71	-0.3004831656	0.12552337	0.204117066
## 72	-0.6217472283	0.37359587	0.760916594
## 73	-0.7148404521	0.42534250	0.587692704
## 74	-0.2736716973	-0.21654820	-0.064072602
## 75	0.2616471875	0.52569153	-0.375791814
## 76	-0.7031636421	-0.21212410	0.184205785
## 77	0.4259891220	-0.20756041	-0.183859506
## 78	0.1534207997	0.07276770	0.029640655
## 79	-0.5363159305	-0.15307912	-0.070950246
## 80	0.3350190034	0.85107024	0.066111114
## 81	-0.4049081907	-0.18855040	0.247600489
## 82	0.3907530118	-0.20552794	-0.099259487
## 83	0.6084468182	0.66276882	0.076478202
## 84	1.1414814164	0.88617702	0.040220101
## 85	0.5003348026	0.40095970	-0.128266231
## 86	0.1222295170	0.40086000	-0.033240314
## 87	-0.5595259481	-0.15469603	-0.657858098

```

## 88 -0.1633343828 -0.22282445 -0.063996092
## 89 -0.6158188116 -1.12693908 0.445965970
## 90 0.3639677031 0.32356598 -0.266883056
## 91 -0.0776519985 0.74583563 0.689480247
## 92 -0.3146249830 -0.87171661 0.501719501
## 93 -0.5280773831 0.34146586 0.078342898
## 94 -0.3949240309 -0.16320876 -0.046207510
## 95 0.2571860166 -0.40213547 0.112379017
## 96 1.7003208877 -2.94676390 -1.058594839
## 97 1.9873942851 0.17642603 0.554999369
## 98 0.2830213414 -0.17314998 -0.083900817
## 99 0.3685569665 -0.19105077 -0.051250402
## 100 0.6065903130 -0.49340761 0.901746589
## 101 -1.9935215752 -1.14236478 -0.124162554
## 102 -0.1927189112 -0.13621111 0.107202134
## 103 -0.3058031426 0.14842006 0.186191916
## 104 -0.3068224231 -0.19018922 0.294267788
## 105 0.0376694600 0.10603791 0.030466947
## 106 1.2004202260 0.35012604 0.430408936
## 107 0.4162225349 -0.22850894 0.077226416
## 108 -0.3010366787 0.12702189 -0.089672025
## 109 -0.2058064932 -0.15631287 -0.076696294
## 110 -0.4066809765 0.13741590 0.238442933
## 111 -0.2704238006 -0.18705929 0.027015529
## 112 1.8010328037 -0.24670776 0.663535696
## 113 0.1909699764 0.10688884 -0.048195644
## 114 -0.1449982851 0.11714335 0.133927338
## 115 0.0097020496 -1.02861486 -0.164182023
## 116 0.0398066964 0.24786659 0.115511203
## 117 -0.6755187287 0.72768949 -0.231209035
## 118 0.2429353351 0.18269689 0.019915353
## 119 0.4554416363 -0.82946182 -0.623637445
## 120 0.5296724555 0.52810692 -0.203702547
## 121 -0.2315616255 0.11832517 0.072427061
## 122 -0.6225769134 -0.37411712 -0.652858571
## 123 -0.2857181484 0.43491691 0.032780079
## 124 -0.3715130180 -0.21279027 -0.028505579
## 125 -0.2203070845 -0.20843752 0.166967572
## 126 -0.2686720442 -0.22930100 0.134945559
## 127 -0.3037082319 0.10439910 0.021922486
## 128 -0.1616240288 0.78273644 -0.086639766
## 129 0.2374639884 0.16451359 -0.158625092
## 130 0.2945634692 0.08299878 0.187836206
## 131 -0.2594982679 -0.22301363 -0.001840435
## 132 -0.4905511416 0.09784380 0.240268603
## 133 0.1109646141 0.76771233 -0.047374155
## 134 0.2647664146 0.36757198 0.028219570
## 135 -0.5456648668 0.29310706 -0.312381820
## 136 -0.3021092512 0.17927705 -0.291229586
## 137 -0.2880344631 -0.16808015 -0.116885402
## 138 -0.0190312678 -0.82881280 0.109707268
## 139 0.0669934180 -0.13516805 -0.182546755
## 140 0.4538242779 0.60807221 -0.362368641
## 141 0.1316524544 0.83184640 0.096116553

```

```

## 142 -0.3224714296 -0.22612973 0.085513662
## 143 0.2077237452 -0.82578562 -0.005185148
## 144 0.2882601370 -0.18344278 -0.193679647
## 145 0.4859300467 -0.66984793 0.155367450
## 146 0.1012382811 0.42496176 -0.290838289
## 147 -0.6741237173 0.50671349 -0.342373105
## 148 -0.3156790988 -0.21576928 0.410490814
## 149 0.5322008381 0.12074107 -0.292627776
## 150 -0.2346858296 -0.22278265 -0.011926061
## 151 -0.1180780318 0.18696636 -0.275829228
## 152 0.0244337343 0.97905957 0.100365173
## 153 0.0839061208 0.40759367 0.026358490
## 154 0.2532018700 0.11546219 -0.440228590
## 155 -0.3011947928 -0.22540589 0.057411714
## 156 -0.0290524597 0.47553138 -0.588827473
## 157 0.5281628656 0.42755390 -0.302341511
## 158 0.2189079037 0.10160772 0.874213970
## 159 -0.1031427198 -0.30212310 -0.088940147
## 160 -0.9120431378 -0.45913493 0.693114623
## 161 -0.1566459981 -0.20493853 0.036319480
## 162 0.1169189470 0.42938274 -0.050300508
## 163 0.3103052412 0.14623935 -0.086914072
## 164 -0.5362151505 0.02600726 0.048180891
## 165 0.2066722487 0.23908267 0.015777024
## 166 0.0819319922 0.48039472 0.082200944
## 167 0.4212675312 0.42439223 -0.063000875
## 168 -0.7276764615 -1.42450764 0.619796432
## 169 -1.7156545419 0.06318158 1.513920149
## 170 0.1139346057 0.10624208 0.149941782
## 171 -0.9921324652 -1.03233531 -0.244789764
## 172 0.9469878124 -0.15889058 0.479326704
## 173 -2.0495450662 0.51790023 -0.330516787
## 174 0.1383527555 0.22150814 -0.063382479
## 175 0.1748825497 0.06249028 -0.137428098
## 176 -1.4886287734 -0.09285025 -0.161477292
## 177 -0.2459708133 -0.20772109 -0.046990106
## 178 -0.1571527212 -0.19114065 0.140501419
## 179 -0.1597784372 0.19916674 -0.107356015
## 180 -0.1930234666 -0.18366772 0.351855031
## 181 -0.0845980597 -0.66973598 -0.232927321
## 182 1.2983503473 -0.68331683 0.478896860
## 183 1.0567421233 0.97292153 -0.200028808
## 184 0.3422127625 0.21814047 0.052602285
## 185 -0.8230630062 -0.20436957 0.175117510
## 186 0.0559634717 0.06255995 -0.050644713
## 187 -0.2362094037 0.51005908 -0.682039560
## 188 0.5413345878 -0.46345163 0.104462940
## 189 -0.1281518971 0.23652308 0.288729074
## 190 0.5828389709 0.12167522 -0.183216436
## 191 -0.0354879214 -0.36472916 0.043996297
## 192 -0.5333889763 0.58054954 0.377026305
## 193 -0.2578763066 0.77227412 -0.391975453
## 194 0.4043906567 0.26099807 -0.381267673
## 195 -0.0009935399 0.39075418 -0.035614077

```

```
## 196 0.1938076348 0.40332734 0.119388709
## 197 0.4116819322 -0.05750868 -0.287258782
## 198 0.2359013218 0.13268758 0.137941247
## 199 -0.8546416298 0.14969497 -0.090368227
## 200 0.0959579417 0.45223389 -0.028134034
```

Means of scores for all the PC's classified by rent status

```
tabmeansPC <- aggregate(rent_amount_pca[,2:8],by=list(Rent=house_data$rent),mean)
tabmeansPC
```

```
##   Rent      PC1      PC2      PC3      PC4      PC5      PC6
## 1 high 2.1915308 0.14258453 -0.3020136 -0.18167517 -0.10121578 0.04112931
## 2 low -0.9617509 -0.06257307 0.1325384 0.07972795 0.04441843 -0.01804955
##
##      PC7
## 1 -0.08067297
## 2 0.03540324
```

```
tabmeansPC <- tabmeansPC[rev(order(tabmeansPC$Rent)),]
tabmeansPC
```

```
##   Rent      PC1      PC2      PC3      PC4      PC5      PC6
## 2 low -0.9617509 -0.06257307 0.1325384 0.07972795 0.04441843 -0.01804955
## 1 high 2.1915308 0.14258453 -0.3020136 -0.18167517 -0.10121578 0.04112931
##
##      PC7
## 2 0.03540324
## 1 -0.08067297
```

```
tabfmeans <- t(tabmeansPC[,1])
tabfmeans
```

```
##           2           1
## PC1 -0.96175094 2.19153084
## PC2 -0.06257307 0.14258453
## PC3 0.13253836 -0.30201363
## PC4 0.07972795 -0.18167517
## PC5 0.04441843 -0.10121578
## PC6 -0.01804955 0.04112931
## PC7 0.03540324 -0.08067297
```

```
colnames(tabfmeans) <- t(as.vector(tabmeansPC[1]$Rent))
tabfmeans
```

```
##           low           high
## PC1 -0.96175094 2.19153084
## PC2 -0.06257307 0.14258453
## PC3 0.13253836 -0.30201363
## PC4 0.07972795 -0.18167517
## PC5 0.04441843 -0.10121578
## PC6 -0.01804955 0.04112931
## PC7 0.03540324 -0.08067297
```

```
# Standard deviations of scores for all the PC's classified by rent status
tabstdsPC <- aggregate(rent_amount_pca[,2:8],by=list(Rent=house_data$rent),sd)
tabfsds <- t(tabstdsPC[,-1])
colnames(tabfsds) <- t(as.vector(tabstdsPC[1]$Rent))
tabfsds
```

```
##           high           low
## PC1 2.0683291 1.2048325
## PC2 1.1876981 0.7754820
## PC3 1.0436622 0.5856605
## PC4 0.8740072 0.5069594
## PC5 0.8239896 0.4392800
## PC6 0.5552749 0.4360745
## PC7 0.5342930 0.2595674
```

```
t.test(PC1~house_data$rent,data=rent_amount_pca)
```

```
##
## Welch Two Sample t-test
##
## data: PC1 by house_data$rent
## t = 11.109, df = 78.444, p-value < 2.2e-16
## alternative hypothesis: true difference in means between group high and group low is not equal to 0
## 95 percent confidence interval:
## 2.588218 3.718345
## sample estimates:
## mean in group high mean in group low
## 2.1915308 -0.9617509
```

```
t.test(PC2~house_data$rent,data=rent_amount_pca)
```

```
##
## Welch Two Sample t-test
##
## data: PC2 by house_data$rent
## t = 1.2382, df = 83.283, p-value = 0.2191
## alternative hypothesis: true difference in means between group high and group low is not equal to 0
## 95 percent confidence interval:
## -0.1243662 0.5346814
## sample estimates:
## mean in group high mean in group low
## 0.14258453 -0.06257307
```

```
t.test(PC3~house_data$rent,data=rent_amount_pca)
```

```
##
## Welch Two Sample t-test
##
## data: PC3 by house_data$rent
## t = -3.0482, df = 77.089, p-value = 0.003153
## alternative hypothesis: true difference in means between group high and group low is not equal to 0
```

```
## 95 percent confidence interval:
## -0.7184235 -0.1506805
## sample estimates:
## mean in group high mean in group low
## -0.3020136 0.1325384
```

```
t.test(PC4~house_data$rent,data=rent_amount_pca)
```

```
##
## Welch Two Sample t-test
##
## data: PC4 by house_data$rent
## t = -2.1805, df = 78.284, p-value = 0.03222
## alternative hypothesis: true difference in means between group high and group low is not equal to 0
## 95 percent confidence interval:
## -0.50005662 -0.02274962
## sample estimates:
## mean in group high mean in group low
## -0.18167517 0.07972795
```

```
t.test(PC5~house_data$rent,data=rent_amount_pca)
```

```
##
## Welch Two Sample t-test
##
## data: PC5 by house_data$rent
## t = -1.3016, df = 75.391, p-value = 0.197
## alternative hypothesis: true difference in means between group high and group low is not equal to 0
## 95 percent confidence interval:
## -0.36850582 0.07723741
## sample estimates:
## mean in group high mean in group low
## -0.10121578 0.04441843
```

```
t.test(PC6~house_data$rent,data=rent_amount_pca)
```

```
##
## Welch Two Sample t-test
##
## data: PC6 by house_data$rent
## t = 0.73843, df = 93.884, p-value = 0.4621
## alternative hypothesis: true difference in means between group high and group low is not equal to 0
## 95 percent confidence interval:
## -0.0999465 0.2183042
## sample estimates:
## mean in group high mean in group low
## 0.04112931 -0.01804955
```

```
t.test(PC7~house_data$rent,data=rent_amount_pca)
```

```
##
```

```
## Welch Two Sample t-test
##
## data: PC7 by house_data$rent
## t = -1.6152, df = 72.733, p-value = 0.1106
## alternative hypothesis: true difference in means between group high and group low is not equal to 0
## 95 percent confidence interval:
## -0.25931118 0.02715875
## sample estimates:
## mean in group high mean in group low
## -0.08067297 0.03540324
```

```
## F ratio tests
var.test(PC1~house_data$rent,data=rent_amount_pca)
```

```
##
## F test to compare two variances
##
## data: PC1 by house_data$rent
## F = 2.947, num df = 60, denom df = 138, p-value = 1.908e-07
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
## 1.949284 4.623972
## sample estimates:
## ratio of variances
## 2.947039
```

```
var.test(PC2~house_data$rent,data=rent_amount_pca)
```

```
##
## F test to compare two variances
##
## data: PC2 by house_data$rent
## F = 2.3457, num df = 60, denom df = 138, p-value = 4.36e-05
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
## 1.551522 3.680424
## sample estimates:
## ratio of variances
## 2.345679
```

```
var.test(PC3~house_data$rent,data=rent_amount_pca)
```

```
##
## F test to compare two variances
##
## data: PC3 by house_data$rent
## F = 3.1756, num df = 60, denom df = 138, p-value = 2.47e-08
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
## 2.100474 4.982615
## sample estimates:
## ratio of variances
## 3.175617
```

```
var.test(PC4~house_data$rent,data=rent_amount_pca)
```

```
##
## F test to compare two variances
##
## data: PC4 by house_data$rent
## F = 2.9722, num df = 60, denom df = 138, p-value = 1.521e-07
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
## 1.965952 4.663509
## sample estimates:
## ratio of variances
## 2.972238
```

```
var.test(PC5~house_data$rent,data=rent_amount_pca)
```

```
##
## F test to compare two variances
##
## data: PC5 by house_data$rent
## F = 3.5185, num df = 60, denom df = 138, p-value = 1.208e-09
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
## 2.327286 5.520644
## sample estimates:
## ratio of variances
## 3.518524
```

```
var.test(PC6~house_data$rent,data=rent_amount_pca)
```

```
##
## F test to compare two variances
##
## data: PC6 by house_data$rent
## F = 1.6214, num df = 60, denom df = 138, p-value = 0.02175
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
## 1.072467 2.544040
## sample estimates:
## ratio of variances
## 1.621417
```

```
var.test(PC7~house_data$rent,data=rent_amount_pca)
```

```
##
## F test to compare two variances
##
## data: PC7 by house_data$rent
## F = 4.237, num df = 60, denom df = 138, p-value = 2.763e-12
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
```



```
## 2.802515 6.647953
## sample estimates:
## ratio of variances
## 4.237002
```

```
# Levene's tests (one-sided)
library(car)
```

```
## Loading required package: carData
```

```
(LTPC1 <- leveneTest(PC1~house_data$rent,data=rent_amount_pca))
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##      Df F value    Pr(>F)
## group  1  20.39 1.083e-05 ***
##      198
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
(p_PC1_1sided <- LTPC1[[3]][1]/2)
```

```
## [1] 5.41647e-06
```

```
(LTPC2 <- leveneTest(PC2~house_data$rent,data=rent_amount_pca))
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##      Df F value    Pr(>F)
## group  1  10.853 0.001169 **
##      198
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
(p_PC2_1sided=LTPC2[[3]][1]/2)
```

```
## [1] 0.0005843576
```

```
(LTPC3 <- leveneTest(PC3~house_data$rent,data=rent_amount_pca))
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##      Df F value    Pr(>F)
## group  1  29.665 1.513e-07 ***
##      198
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
(p_PC3_1sided <- LTPC3[[3]][1]/2)
```

```
## [1] 7.56314e-08
```

```
(LTPC4 <- leveneTest(PC4~house_data$rent,data=rent_amount_pca))
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##      Df F value    Pr(>F)
## group  1  12.721 0.0004534 ***
##      198
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
(p_PC4_1sided <- LTPC4[[3]][1]/2)
```

```
## [1] 0.0002267165
```

```
(LTPC5 <- leveneTest(PC5~house_data$rent,data=rent_amount_pca))
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##      Df F value    Pr(>F)
## group  1  26.113 7.557e-07 ***
##      198
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
(p_PC5_1sided <- LTPC5[[3]][1]/2)
```

```
## [1] 3.778356e-07
```

```
(LTPC6 <- leveneTest(PC6~house_data$rent,data=rent_amount_pca))
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##      Df F value    Pr(>F)
## group  1  8.1872 0.004672 **
##      198
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
(p_PC6_1sided <- LTPC6[[3]][1]/2)
```

```
## [1] 0.002335952
```

```
(LTPC7 <- leveneTest(PC7~house_data$rent,data=rent_amount_pca))
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

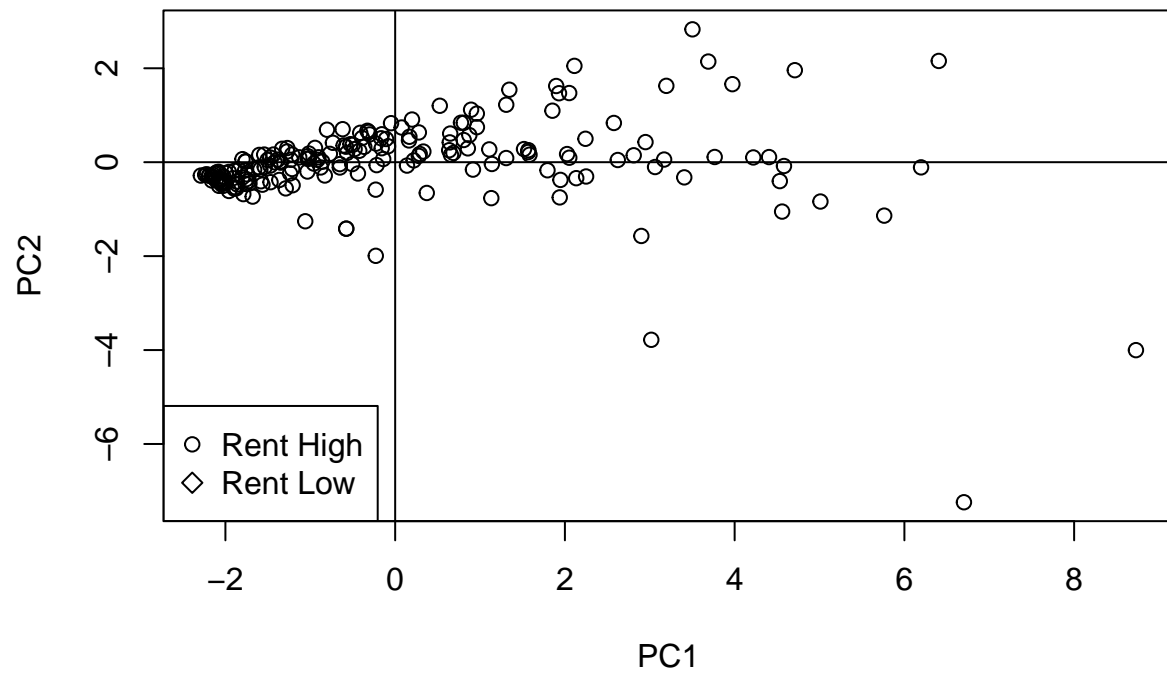
```
## Levene's Test for Homogeneity of Variance (center = median)
##      Df F value    Pr(>F)
## group  1 24.227 1.799e-06 ***
##      198
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
(p_PC7_1sided <- LTPC7[[3]][1]/2)
```

```
## [1] 8.994908e-07
```

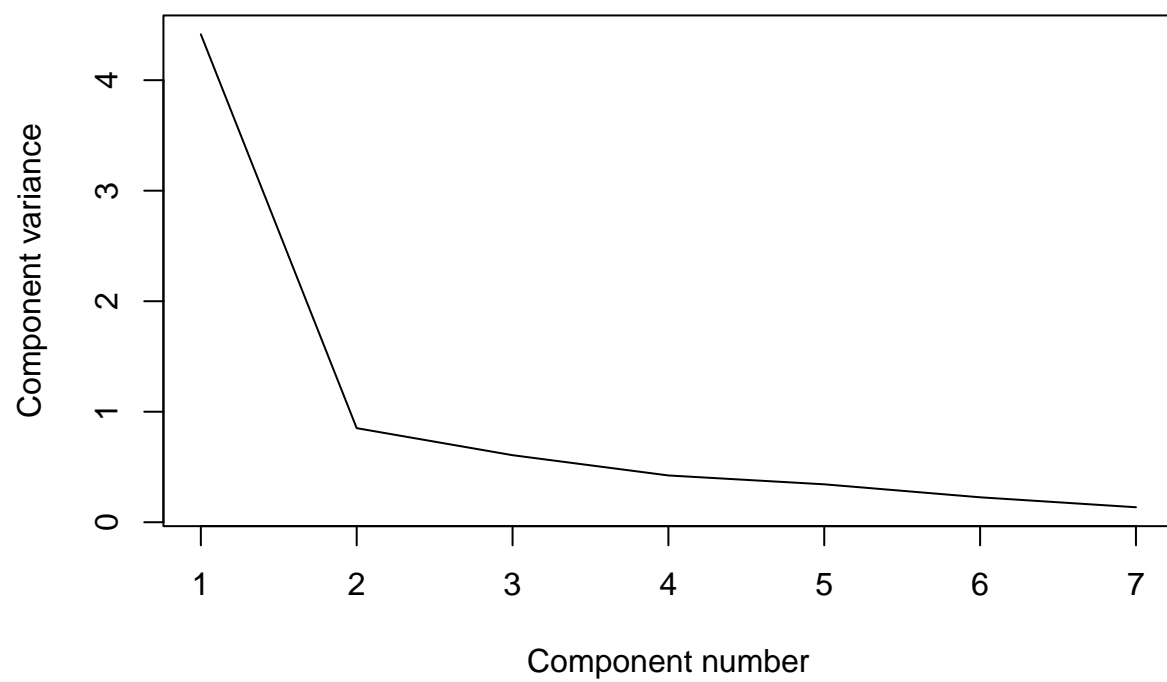
```
# Plotting the scores for the first and second components
plot(rent_amount_pca$PC1, rent_amount_pca$PC2,pch=ifelse(rent_amount_pca$rent == "high",1,5),xlab="PC1"
abline(h=0)
abline(v=0)
legend("bottomleft", legend=c("Rent High","Rent Low"), pch=c(1,5))
```

rent prices against values for PC1 & PC2



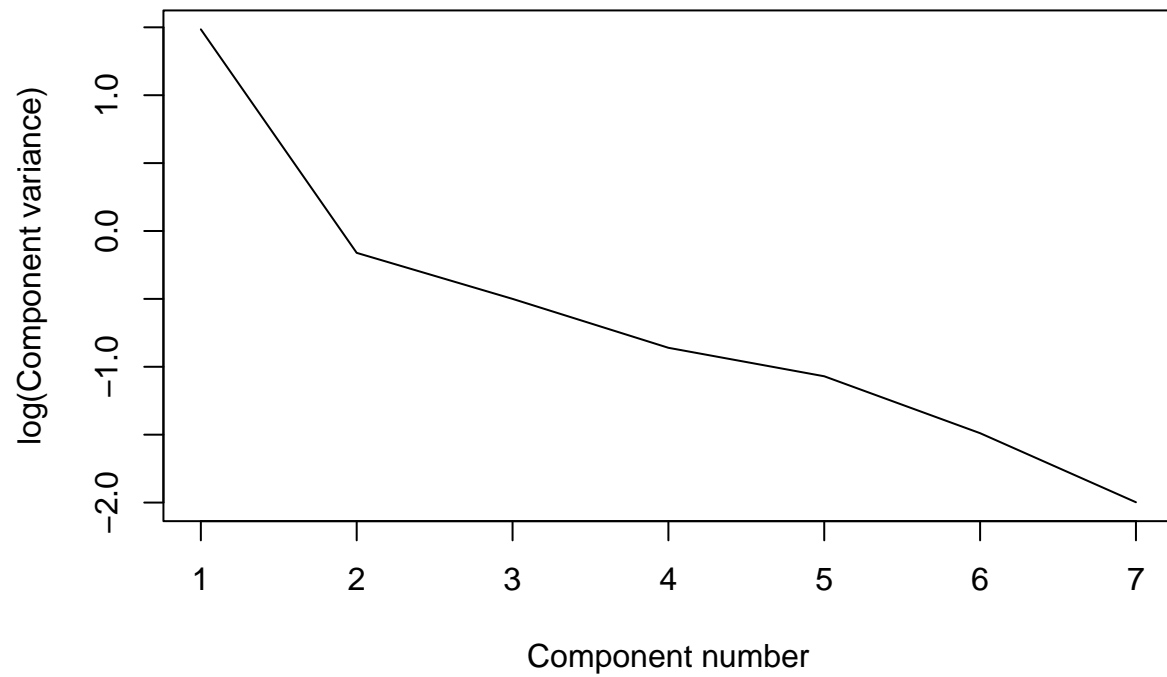
```
plot(eigen_rent, xlab = "Component number", ylab = "Component variance", type = "l", main = "Scree diag")
```

Scree diagram

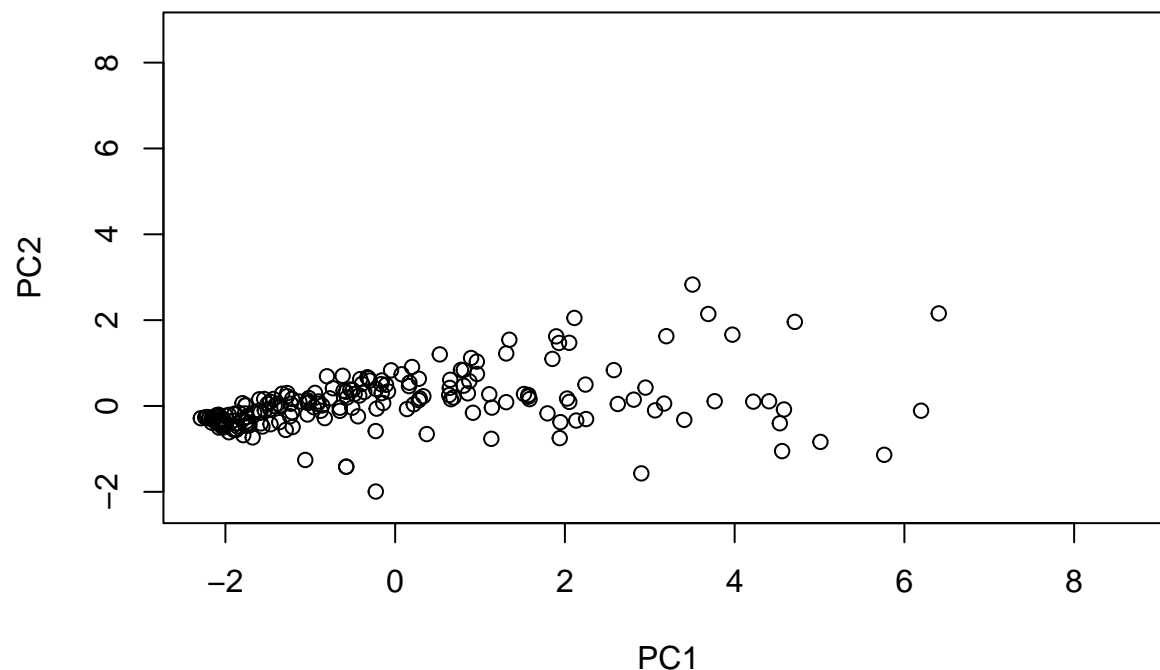


```
plot(log(eigen_rent), xlab = "Component number",ylab = "log(Component variance)", type="l",main = "Log(
```

Log(eigenvalue) diagram



```
xlim <- range(rent_pca$x[,1])  
plot(rent_pca$x,xlim=xlim,ylim=xlim)
```



```
rent_pca$rotation[,1]
```

```
##          area          rooms    bathroom parking spaces          hoa
##    0.4336965    0.3704037    0.4227960    0.3939576    0.3144122
##  property tax fire insurance
##    0.2933799    0.3947438
```

```
rent_pca$rotation
```

```
##          PC1          PC2          PC3          PC4          PC5
## area      0.4336965  0.17091900  0.11991336 -0.14262889 -0.35241854
## rooms     0.3704037  0.38742007  0.06788605  0.73084114  0.02399273
## bathroom  0.4227960  0.15626819  0.06607019  0.09708057  0.28600555
## parking spaces 0.3939576  0.08267643  0.27820446 -0.52057449  0.56700711
## hoa       0.3144122 -0.55087277 -0.64250123  0.17481916  0.31296716
## property tax 0.2933799 -0.68199336  0.54402406  0.12426281 -0.30542261
## fire insurance 0.3947438  0.14443506 -0.43643932 -0.34511274 -0.52980928
##          PC6          PC7
## area      -0.04502695  0.78850847
## rooms      0.39473933 -0.13256958
## bathroom  -0.81830269 -0.17780729
## parking spaces 0.39883150 -0.09488344
## hoa        0.08754800  0.22068420
## property tax 0.01872523 -0.20922784
## fire insurance 0.07383965 -0.47705723
```

```

#get the original value of the data based on PCA
center <- rent_pca$center
scale <- rent_pca$scale
new_rent <- as.matrix(house_data[, -11])
new_rent

```

```

##      area rooms bathroom parking spaces animal      furniture      hoa
## [1,] "120" "3"  "4"      "3"          "accept"    "not furnished" "1350"
## [2,] " 45" "1"  "1"      "1"          "not accept" "furnished"     "3000"
## [3,] " 50" "2"  "1"      "1"          "accept"     "not furnished" " 226"
## [4,] " 35" "1"  "1"      "0"          "accept"     "not furnished" " 260"
## [5,] "204" "4"  "4"      "2"          "accept"     "not furnished" "   0"
## [6,] "177" "3"  "3"      "4"          "accept"     "not furnished" "2700"
## [7,] " 15" "1"  "1"      "0"          "not accept" "not furnished" "   0"
## [8,] " 70" "2"  "2"      "1"          "accept"     "furnished"     "1800"
## [9,] "180" "3"  "3"      "2"          "accept"     "not furnished" " 700"
## [10,] "180" "4"  "4"      "2"          "accept"     "not furnished" "2600"
## [11,] " 70" "2"  "2"      "1"          "accept"     "not furnished" " 650"
## [12,] "200" "3"  "4"      "3"          "accept"     "furnished"     "1800"
## [13,] "100" "2"  "2"      "0"          "accept"     "not furnished" "   0"
## [14,] " 21" "1"  "1"      "0"          "accept"     "not furnished" " 519"
## [15,] " 65" "2"  "1"      "1"          "accept"     "not furnished" " 360"
## [16,] "100" "3"  "2"      "0"          "not accept" "not furnished" " 815"
## [17,] " 64" "1"  "2"      "1"          "accept"     "furnished"     " 650"
## [18,] " 50" "1"  "1"      "0"          "accept"     "not furnished" " 985"
## [19,] "137" "3"  "3"      "1"          "accept"     "furnished"     "1200"
## [20,] " 43" "1"  "1"      "0"          "not accept" "not furnished" " 350"
## [21,] " 49" "2"  "1"      "1"          "accept"     "not furnished" "   0"
## [22,] "101" "3"  "2"      "2"          "accept"     "furnished"     " 541"
## [23,] "330" "4"  "5"      "4"          "accept"     "not furnished" "   0"
## [24,] "320" "8"  "4"      "0"          "accept"     "not furnished" " 450"
## [25,] " 50" "3"  "2"      "2"          "accept"     "not furnished" " 150"
## [26,] " 55" "1"  "1"      "0"          "accept"     "not furnished" " 700"
## [27,] " 30" "1"  "1"      "0"          "not accept" "furnished"     " 260"
## [28,] " 40" "1"  "1"      "1"          "not accept" "furnished"     "   0"
## [29,] " 73" "2"  "2"      "1"          "accept"     "not furnished" " 700"
## [30,] "240" "3"  "4"      "2"          "accept"     "not furnished" "   1"
## [31,] " 38" "1"  "1"      "0"          "not accept" "not furnished" " 700"
## [32,] "144" "4"  "2"      "2"          "accept"     "not furnished" "1820"
## [33,] "161" "3"  "3"      "3"          "accept"     "not furnished" "1300"
## [34,] " 85" "2"  "1"      "1"          "accept"     "not furnished" " 400"
## [35,] "162" "4"  "5"      "3"          "accept"     "not furnished" "1720"
## [36,] "170" "3"  "5"      "3"          "not accept" "not furnished" "2091"
## [37,] "620" "4"  "6"      "4"          "accept"     "not furnished" "8133"
## [38,] "120" "3"  "2"      "1"          "accept"     "not furnished" "   0"
## [39,] " 23" "1"  "1"      "0"          "not accept" "not furnished" " 550"
## [40,] "135" "4"  "2"      "1"          "accept"     "not furnished" " 600"
## [41,] " 48" "1"  "1"      "0"          "not accept" "not furnished" "   0"
## [42,] "209" "4"  "4"      "4"          "accept"     "not furnished" "2140"
## [43,] "130" "3"  "3"      "2"          "accept"     "not furnished" "1200"
## [44,] " 83" "3"  "3"      "2"          "accept"     "furnished"     " 450"
## [45,] "190" "4"  "3"      "1"          "accept"     "not furnished" "1570"
## [46,] "400" "3"  "3"      "4"          "accept"     "not furnished" "   0"

```


##	[47,]	"250"	"4"	"4"	"4"	"accept"	"not furnished"	"2600"
##	[48,]	" 71"	"3"	"1"	"0"	"accept"	"not furnished"	" 320"
##	[49,]	"303"	"3"	"4"	"4"	"accept"	"furnished"	"4540"
##	[50,]	" 50"	"1"	"1"	"0"	"accept"	"furnished"	" 200"
##	[51,]	" 89"	"3"	"2"	"1"	"accept"	"not furnished"	" 860"
##	[52,]	" 68"	"2"	"2"	"1"	"accept"	"not furnished"	" 400"
##	[53,]	" 65"	"3"	"2"	"1"	"accept"	"not furnished"	" 512"
##	[54,]	"305"	"3"	"3"	"3"	"accept"	"not furnished"	"3100"
##	[55,]	"300"	"4"	"5"	"5"	"not accept"	"not furnished"	"3500"
##	[56,]	" 63"	"2"	"1"	"0"	"accept"	"not furnished"	" 600"
##	[57,]	" 35"	"1"	"1"	"0"	"accept"	"not furnished"	" 270"
##	[58,]	" 50"	"1"	"1"	"0"	"not accept"	"not furnished"	" 0"
##	[59,]	" 70"	"2"	"1"	"1"	"accept"	"not furnished"	" 729"
##	[60,]	" 76"	"2"	"1"	"1"	"accept"	"not furnished"	" 0"
##	[61,]	" 48"	"1"	"1"	"0"	"accept"	"not furnished"	" 309"
##	[62,]	" 83"	"2"	"3"	"2"	"accept"	"not furnished"	"1555"
##	[63,]	"100"	"3"	"3"	"0"	"accept"	"not furnished"	" 820"
##	[64,]	" 40"	"1"	"1"	"0"	"not accept"	"not furnished"	" 0"
##	[65,]	"170"	"3"	"3"	"2"	"accept"	"not furnished"	" 0"
##	[66,]	"120"	"3"	"2"	"1"	"accept"	"not furnished"	" 300"
##	[67,]	"165"	"3"	"3"	"2"	"accept"	"furnished"	"1150"
##	[68,]	" 44"	"1"	"1"	"1"	"accept"	"not furnished"	" 390"
##	[69,]	" 22"	"1"	"1"	"0"	"not accept"	"furnished"	" 400"
##	[70,]	" 44"	"1"	"1"	"0"	"accept"	"furnished"	" 270"
##	[71,]	" 70"	"2"	"1"	"0"	"not accept"	"not furnished"	" 350"
##	[72,]	"288"	"3"	"2"	"2"	"accept"	"not furnished"	" 0"
##	[73,]	"250"	"4"	"2"	"1"	"accept"	"not furnished"	" 0"
##	[74,]	" 25"	"1"	"1"	"0"	"not accept"	"not furnished"	" 0"
##	[75,]	" 95"	"3"	"2"	"2"	"accept"	"furnished"	"1000"
##	[76,]	" 90"	"1"	"1"	"0"	"accept"	"not furnished"	" 0"
##	[77,]	" 40"	"2"	"2"	"1"	"not accept"	"not furnished"	" 210"
##	[78,]	" 40"	"1"	"1"	"1"	"not accept"	"not furnished"	" 225"
##	[79,]	" 51"	"1"	"1"	"0"	"not accept"	"not furnished"	" 429"
##	[80,]	"150"	"4"	"2"	"2"	"accept"	"not furnished"	"1357"
##	[81,]	"130"	"3"	"2"	"0"	"accept"	"not furnished"	" 0"
##	[82,]	" 52"	"2"	"2"	"1"	"accept"	"not furnished"	" 285"
##	[83,]	"150"	"3"	"2"	"3"	"accept"	"not furnished"	" 0"
##	[84,]	" 80"	"1"	"1"	"4"	"accept"	"not furnished"	" 0"
##	[85,]	" 95"	"3"	"2"	"2"	"not accept"	"not furnished"	" 150"
##	[86,]	" 50"	"2"	"1"	"1"	"accept"	"not furnished"	" 158"
##	[87,]	"112"	"1"	"2"	"2"	"accept"	"furnished"	" 0"
##	[88,]	" 15"	"1"	"1"	"0"	"accept"	"not furnished"	" 0"
##	[89,]	"190"	"2"	"3"	"0"	"accept"	"not furnished"	" 0"
##	[90,]	"148"	"4"	"3"	"2"	"accept"	"not furnished"	"1886"
##	[91,]	"250"	"4"	"2"	"2"	"accept"	"furnished"	" 600"
##	[92,]	"120"	"1"	"2"	"0"	"accept"	"not furnished"	" 0"
##	[93,]	"550"	"3"	"5"	"8"	"accept"	"not furnished"	" 0"
##	[94,]	" 40"	"1"	"1"	"0"	"not accept"	"not furnished"	" 450"
##	[95,]	"137"	"3"	"3"	"1"	"accept"	"furnished"	"1180"
##	[96,]	" 43"	"2"	"7"	"2"	"accept"	"not furnished"	" 257"
##	[97,]	"147"	"2"	"3"	"4"	"accept"	"not furnished"	"2600"
##	[98,]	" 65"	"2"	"2"	"1"	"accept"	"not furnished"	" 532"
##	[99,]	" 60"	"2"	"2"	"1"	"accept"	"not furnished"	" 440"
##	[100,]	"100"	"1"	"2"	"0"	"not accept"	"not furnished"	"4320"

## [101,]	"362"	"4"	"5"	"0"	"not accept"	"not furnished"	"4138"
## [102,]	" 40"	"1"	"1"	"0"	"accept"	"not furnished"	" 928"
## [103,]	" 68"	"2"	"1"	"0"	"accept"	"not furnished"	" 593"
## [104,]	" 68"	"1"	"1"	"0"	"accept"	"not furnished"	" 500"
## [105,]	" 52"	"1"	"1"	"1"	"accept"	"not furnished"	" 470"
## [106,]	"268"	"4"	"4"	"4"	"accept"	"furnished"	"3317"
## [107,]	"250"	"3"	"4"	"3"	"accept"	"furnished"	"2900"
## [108,]	" 39"	"2"	"1"	"0"	"accept"	"not furnished"	" 120"
## [109,]	" 20"	"1"	"1"	"0"	"accept"	"furnished"	" 602"
## [110,]	" 84"	"2"	"1"	"0"	"accept"	"not furnished"	" 420"
## [111,]	" 36"	"1"	"1"	"0"	"accept"	"furnished"	" 345"
## [112,]	"387"	"4"	"6"	"6"	"accept"	"not furnished"	"4246"
## [113,]	" 30"	"1"	"1"	"1"	"accept"	"not furnished"	" 500"
## [114,]	" 80"	"1"	"1"	"1"	"accept"	"not furnished"	" 550"
## [115,]	" 72"	"2"	"3"	"0"	"accept"	"furnished"	" 826"
## [116,]	"125"	"3"	"2"	"1"	"accept"	"furnished"	"1460"
## [117,]	"200"	"3"	"2"	"2"	"accept"	"not furnished"	"2573"
## [118,]	" 95"	"3"	"2"	"1"	"accept"	"not furnished"	" 850"
## [119,]	"201"	"3"	"5"	"3"	"not accept"	"furnished"	"2550"
## [120,]	" 90"	"3"	"2"	"2"	"accept"	"not furnished"	"1300"
## [121,]	" 50"	"2"	"1"	"0"	"not accept"	"not furnished"	" 200"
## [122,]	" 77"	"1"	"2"	"1"	"accept"	"furnished"	" 750"
## [123,]	" 95"	"2"	"1"	"1"	"accept"	"not furnished"	" 300"
## [124,]	" 38"	"1"	"1"	"0"	"accept"	"not furnished"	" 0"
## [125,]	" 45"	"1"	"1"	"0"	"accept"	"not furnished"	" 300"
## [126,]	" 45"	"1"	"1"	"0"	"accept"	"not furnished"	" 50"
## [127,]	" 50"	"2"	"1"	"0"	"not accept"	"not furnished"	" 0"
## [128,]	" 86"	"3"	"1"	"1"	"accept"	"not furnished"	" 460"
## [129,]	" 75"	"3"	"2"	"1"	"not accept"	"not furnished"	" 550"
## [130,]	" 45"	"1"	"1"	"1"	"accept"	"not furnished"	" 521"
## [131,]	" 30"	"1"	"1"	"0"	"accept"	"not furnished"	" 0"
## [132,]	" 90"	"2"	"1"	"0"	"accept"	"not furnished"	" 0"
## [133,]	" 65"	"3"	"1"	"1"	"accept"	"not furnished"	" 510"
## [134,]	" 45"	"1"	"1"	"1"	"not accept"	"furnished"	"3000"
## [135,]	"312"	"4"	"4"	"3"	"accept"	"not furnished"	"4000"
## [136,]	" 51"	"1"	"1"	"1"	"accept"	"furnished"	" 714"
## [137,]	" 23"	"1"	"1"	"0"	"accept"	"not furnished"	" 399"
## [138,]	" 54"	"1"	"2"	"0"	"accept"	"not furnished"	" 245"
## [139,]	" 75"	"2"	"2"	"1"	"accept"	"not furnished"	" 700"
## [140,]	"170"	"4"	"3"	"3"	"accept"	"not furnished"	"1800"
## [141,]	"170"	"4"	"2"	"2"	"accept"	"not furnished"	"1100"
## [142,]	" 45"	"1"	"1"	"0"	"not accept"	"not furnished"	" 0"
## [143,]	"200"	"3"	"4"	"2"	"accept"	"not furnished"	" 0"
## [144,]	" 52"	"2"	"2"	"1"	"not accept"	"not furnished"	" 360"
## [145,]	"200"	"3"	"4"	"2"	"accept"	"not furnished"	"1730"
## [146,]	"115"	"3"	"2"	"2"	"accept"	"not furnished"	" 0"
## [147,]	"360"	"5"	"4"	"4"	"accept"	"not furnished"	" 0"
## [148,]	" 80"	"1"	"1"	"0"	"accept"	"not furnished"	" 350"
## [149,]	" 64"	"2"	"2"	"2"	"accept"	"not furnished"	" 450"
## [150,]	" 27"	"1"	"1"	"0"	"not accept"	"not furnished"	" 0"
## [151,]	" 37"	"1"	"1"	"1"	"accept"	"furnished"	" 890"
## [152,]	"160"	"2"	"1"	"3"	"accept"	"not furnished"	" 0"
## [153,]	" 60"	"2"	"1"	"1"	"accept"	"not furnished"	" 250"
## [154,]	" 72"	"2"	"2"	"2"	"accept"	"not furnished"	" 150"

##	[155,]	" 40"	"1"	"1"	"0"	"not accept"	"not furnished"	" 0"
##	[156,]	"190"	"3"	"3"	"3"	"accept"	"not furnished"	"3000"
##	[157,]	" 76"	"3"	"2"	"2"	"accept"	"not furnished"	" 250"
##	[158,]	"365"	"4"	"4"	"3"	"accept"	"furnished"	"3500"
##	[159,]	"154"	"3"	"3"	"1"	"not accept"	"not furnished"	"1700"
##	[160,]	"360"	"5"	"4"	"1"	"accept"	"not furnished"	" 0"
##	[161,]	" 26"	"1"	"1"	"0"	"accept"	"not furnished"	" 250"
##	[162,]	" 50"	"2"	"1"	"1"	"accept"	"not furnished"	" 407"
##	[163,]	" 75"	"3"	"2"	"1"	"accept"	"not furnished"	" 487"
##	[164,]	"250"	"3"	"3"	"2"	"accept"	"not furnished"	"1900"
##	[165,]	" 40"	"1"	"1"	"1"	"accept"	"not furnished"	"1840"
##	[166,]	" 70"	"2"	"1"	"1"	"not accept"	"not furnished"	" 960"
##	[167,]	"110"	"3"	"2"	"2"	"accept"	"not furnished"	" 380"
##	[168,]	"470"	"4"	"6"	"3"	"accept"	"not furnished"	" 0"
##	[169,]	"750"	"4"	"5"	"6"	"accept"	"not furnished"	" 0"
##	[170,]	" 58"	"1"	"1"	"1"	"accept"	"not furnished"	" 610"
##	[171,]	"450"	"4"	"6"	"4"	"accept"	"not furnished"	" 0"
##	[172,]	"187"	"2"	"3"	"3"	"accept"	"not furnished"	"1600"
##	[173,]	"250"	"3"	"2"	"1"	"accept"	"not furnished"	"3200"
##	[174,]	" 96"	"3"	"2"	"1"	"accept"	"not furnished"	"1122"
##	[175,]	" 20"	"1"	"1"	"1"	"not accept"	"not furnished"	" 0"
##	[176,]	"310"	"3"	"3"	"2"	"not accept"	"not furnished"	" 0"
##	[177,]	" 25"	"1"	"1"	"0"	"accept"	"not furnished"	" 105"
##	[178,]	" 38"	"1"	"1"	"0"	"not accept"	"not furnished"	" 457"
##	[179,]	"117"	"3"	"2"	"1"	"accept"	"not furnished"	" 706"
##	[180,]	" 64"	"1"	"1"	"0"	"accept"	"not furnished"	" 680"
##	[181,]	" 89"	"3"	"3"	"0"	"accept"	"furnished"	" 990"
##	[182,]	"253"	"4"	"5"	"3"	"accept"	"not furnished"	"2100"
##	[183,]	"213"	"4"	"3"	"5"	"accept"	"not furnished"	" 0"
##	[184,]	" 90"	"3"	"2"	"1"	"accept"	"not furnished"	"1300"
##	[185,]	"100"	"1"	"1"	"0"	"accept"	"furnished"	" 0"
##	[186,]	" 40"	"1"	"1"	"1"	"accept"	"not furnished"	" 0"
##	[187,]	"285"	"4"	"4"	"4"	"accept"	"not furnished"	"3100"
##	[188,]	"140"	"2"	"3"	"2"	"accept"	"furnished"	"1000"
##	[189,]	"128"	"4"	"2"	"0"	"not accept"	"not furnished"	"1100"
##	[190,]	" 70"	"2"	"2"	"2"	"accept"	"not furnished"	" 610"
##	[191,]	"157"	"3"	"3"	"1"	"accept"	"not furnished"	"1325"
##	[192,]	"220"	"4"	"2"	"1"	"not accept"	"not furnished"	"1340"
##	[193,]	"240"	"4"	"3"	"3"	"accept"	"not furnished"	"2850"
##	[194,]	"160"	"3"	"3"	"3"	"not accept"	"not furnished"	"1509"
##	[195,]	" 60"	"2"	"1"	"1"	"not accept"	"not furnished"	" 0"
##	[196,]	" 60"	"2"	"1"	"1"	"accept"	"not furnished"	" 345"
##	[197,]	"301"	"4"	"5"	"4"	"accept"	"furnished"	"4265"
##	[198,]	"105"	"3"	"2"	"1"	"not accept"	"not furnished"	" 500"
##	[199,]	" 90"	"2"	"1"	"0"	"accept"	"furnished"	" 1"
##	[200,]	" 55"	"2"	"1"	"1"	"accept"	"not furnished"	" 635"
##		rent	amount	property	tax	fire	insurance	
##	[1,]	" 5600"		" 560"		" 71"		
##	[2,]	" 5520"		" 0"		" 70"		
##	[3,]	" 750"		" 0"		" 10"		
##	[4,]	" 1400"		" 0"		" 18"		
##	[5,]	" 3440"		" 100"		" 62"		
##	[6,]	" 6900"		" 509"		" 89"		
##	[7,]	" 1200"		" 0"		" 16"		

##	[8,]	" 4200"	" 250"	" 55"
##	[9,]	" 2700"	" 175"	" 40"
##	[10,]	" 2000"	" 584"	" 26"
##	[11,]	" 1000"	" 45"	" 13"
##	[12,]	" 5058"	" 1200"	" 65"
##	[13,]	" 2500"	" 274"	" 38"
##	[14,]	" 1200"	" 0"	" 16"
##	[15,]	" 1066"	" 15"	" 16"
##	[16,]	" 808"	" 81"	" 11"
##	[17,]	" 8900"	" 154"	" 64"
##	[18,]	" 2308"	" 138"	" 30"
##	[19,]	" 5500"	" 303"	" 70"
##	[20,]	" 1080"	" 57"	" 14"
##	[21,]	" 1000"	" 0"	" 13"
##	[22,]	" 4200"	" 142"	" 62"
##	[23,]	" 6500"	" 750"	" 98"
##	[24,]	"10500"	" 350"	"158"
##	[25,]	" 2800"	" 30"	" 43"
##	[26,]	" 1020"	" 29"	" 15"
##	[27,]	" 1490"	" 15"	" 22"
##	[28,]	" 970"	" 0"	" 13"
##	[29,]	" 1250"	" 150"	" 16"
##	[30,]	" 8200"	" 625"	"124"
##	[31,]	" 1650"	" 0"	" 25"
##	[32,]	" 4800"	" 667"	" 61"
##	[33,]	" 4500"	" 640"	" 58"
##	[34,]	" 1600"	" 0"	" 21"
##	[35,]	" 3700"	" 525"	" 47"
##	[36,]	" 4500"	" 951"	" 19"
##	[37,]	"15000"	" 4520"	"191"
##	[38,]	" 1020"	" 25"	" 15"
##	[39,]	" 1500"	" 0"	" 20"
##	[40,]	" 2200"	" 138"	" 30"
##	[41,]	" 1150"	" 34"	" 18"
##	[42,]	" 7300"	" 587"	" 98"
##	[43,]	" 3500"	" 342"	" 45"
##	[44,]	" 3655"	" 142"	" 49"
##	[45,]	" 3500"	" 506"	" 46"
##	[46,]	" 4250"	" 160"	" 64"
##	[47,]	"15000"	" 1250"	"191"
##	[48,]	" 1250"	" 46"	" 19"
##	[49,]	"18000"	" 0"	"229"
##	[50,]	" 1300"	" 12"	" 19"
##	[51,]	" 1800"	" 70"	" 23"
##	[52,]	" 1800"	" 82"	" 27"
##	[53,]	" 1860"	" 135"	" 24"
##	[54,]	" 5500"	" 2167"	" 70"
##	[55,]	" 1990"	"10830"	" 26"
##	[56,]	" 1110"	" 0"	" 15"
##	[57,]	" 1300"	" 0"	" 17"
##	[58,]	" 1200"	" 0"	" 19"
##	[59,]	" 900"	" 122"	" 12"
##	[60,]	" 2000"	" 45"	" 31"
##	[61,]	" 700"	" 28"	" 11"

##	[62,]	" 3200"	" 735"	" 41"
##	[63,]	" 6330"	" 148"	" 81"
##	[64,]	" 2040"	" 70"	" 31"
##	[65,]	" 8000"	" 209"	"143"
##	[66,]	" 2100"	" 149"	" 28"
##	[67,]	" 2710"	" 84"	" 35"
##	[68,]	" 1200"	" 74"	" 16"
##	[69,]	" 1956"	" 50"	" 25"
##	[70,]	" 550"	" 6"	" 9"
##	[71,]	" 1050"	" 116"	" 14"
##	[72,]	" 3700"	" 457"	" 56"
##	[73,]	" 2700"	" 209"	" 48"
##	[74,]	" 1200"	" 35"	" 16"
##	[75,]	" 4500"	" 325"	" 58"
##	[76,]	" 2200"	" 109"	" 34"
##	[77,]	" 900"	" 0"	" 12"
##	[78,]	" 1080"	" 65"	" 14"
##	[79,]	" 3250"	" 37"	" 42"
##	[80,]	" 3700"	" 211"	" 50"
##	[81,]	" 1400"	" 10"	" 22"
##	[82,]	" 900"	" 59"	" 12"
##	[83,]	" 2430"	" 38"	" 32"
##	[84,]	" 1000"	" 75"	" 16"
##	[85,]	" 1650"	" 145"	" 22"
##	[86,]	" 1140"	" 0"	" 16"
##	[87,]	" 8389"	" 0"	"107"
##	[88,]	" 700"	" 0"	" 10"
##	[89,]	" 2500"	" 14"	" 41"
##	[90,]	" 5500"	" 399"	" 74"
##	[91,]	" 3500"	" 0"	" 47"
##	[92,]	" 761"	" 129"	" 10"
##	[93,]	"10000"	" 3825"	"151"
##	[94,]	" 2390"	" 114"	" 31"
##	[95,]	" 2900"	" 214"	" 38"
##	[96,]	" 2270"	" 0"	" 29"
##	[97,]	" 1200"	" 375"	" 16"
##	[98,]	" 1820"	" 49"	" 24"
##	[99,]	" 1250"	" 38"	" 16"
##	[100,]	" 3100"	" 0"	" 40"
##	[101,]	"15000"	" 1975"	"200"
##	[102,]	" 1950"	" 61"	" 26"
##	[103,]	" 1000"	" 400"	" 13"
##	[104,]	" 1500"	" 0"	" 20"
##	[105,]	" 2000"	" 89"	" 26"
##	[106,]	" 6000"	" 899"	" 77"
##	[107,]	" 9800"	" 125"	"125"
##	[108,]	" 1450"	" 43"	" 19"
##	[109,]	" 1800"	" 130"	" 23"
##	[110,]	" 1700"	" 85"	" 22"
##	[111,]	" 1600"	" 30"	" 21"
##	[112,]	" 6647"	" 2235"	" 85"
##	[113,]	" 1560"	" 61"	" 20"
##	[114,]	" 2800"	" 100"	" 36"
##	[115,]	" 2400"	" 220"	" 32"

## [116,]	" 3500"	" 295"	" 46"
## [117,]	"10000"	" 1121"	"127"
## [118,]	" 2300"	" 0"	" 30"
## [119,]	"10000"	" 795"	"127"
## [120,]	" 3500"	" 250"	" 45"
## [121,]	" 950"	" 0"	" 13"
## [122,]	" 7740"	" 209"	" 99"
## [123,]	" 2800"	" 69"	" 41"
## [124,]	" 1680"	" 0"	" 22"
## [125,]	" 800"	" 108"	" 11"
## [126,]	" 640"	" 84"	" 10"
## [127,]	" 750"	" 59"	" 13"
## [128,]	" 3150"	" 0"	" 40"
## [129,]	" 2125"	" 67"	" 27"
## [130,]	" 600"	" 38"	" 8"
## [131,]	" 1000"	" 50"	" 13"
## [132,]	" 1199"	" 42"	" 19"
## [133,]	" 1800"	" 0"	" 23"
## [134,]	" 5520"	" 0"	" 70"
## [135,]	"15000"	" 1084"	"194"
## [136,]	" 4650"	" 190"	" 59"
## [137,]	" 2100"	" 42"	" 27"
## [138,]	" 720"	" 46"	" 10"
## [139,]	" 3160"	" 165"	" 41"
## [140,]	" 6000"	" 881"	" 80"
## [141,]	" 4000"	" 297"	" 54"
## [142,]	" 950"	" 30"	" 15"
## [143,]	" 3250"	" 292"	" 49"
## [144,]	" 1650"	" 86"	" 22"
## [145,]	" 5000"	" 306"	" 65"
## [146,]	" 3000"	" 71"	" 50"
## [147,]	" 9000"	" 600"	"148"
## [148,]	" 1000"	" 0"	" 14"
## [149,]	" 2800"	" 30"	" 36"
## [150,]	" 835"	" 0"	" 13"
## [151,]	" 4150"	" 110"	" 53"
## [152,]	" 4000"	" 0"	" 61"
## [153,]	" 1300"	" 30"	" 18"
## [154,]	" 3700"	" 184"	" 47"
## [155,]	" 880"	" 48"	" 14"
## [156,]	"13000"	" 84"	"165"
## [157,]	" 2100"	" 0"	" 31"
## [158,]	" 8800"	" 459"	"118"
## [159,]	" 6500"	" 0"	" 83"
## [160,]	" 4600"	" 158"	" 76"
## [161,]	" 800"	" 12"	" 12"
## [162,]	" 1715"	" 12"	" 22"
## [163,]	" 1280"	" 109"	" 18"
## [164,]	" 9500"	" 500"	"121"
## [165,]	" 2990"	" 409"	" 38"
## [166,]	" 2600"	" 0"	" 33"
## [167,]	" 2150"	" 192"	" 29"
## [168,]	" 7200"	" 1134"	"109"
## [169,]	"13000"	" 1500"	"196"

```
## [170,] " 1600"      "   89"      "  21"
## [171,] "11600"      " 1042"     "175"
## [172,] " 3100"      "  292"     " 40"
## [173,] " 8500"      " 5000"     "108"
## [174,] " 3050"      "  231"     " 39"
## [175,] " 1000"      "    0"     " 14"
## [176,] "10400"      "    0"     "157"
## [177,] " 1330"      "    0"     " 17"
## [178,] " 1050"      "    0"     " 14"
## [179,] " 3400"      "  217"     " 50"
## [180,] " 1100"      "   50"     " 14"
## [181,] " 3550"      "   84"     " 45"
## [182,] " 4100"      "    0"     " 52"
## [183,] " 3900"      "  298"     " 59"
## [184,] " 2050"      "  230"     " 27"
## [185,] " 2720"      "  138"     " 41"
## [186,] " 1200"      "    0"     " 19"
## [187,] "15000"      "  973"     "191"
## [188,] " 3000"      "  113"     " 44"
## [189,] " 2000"      "   10"     " 26"
## [190,] " 2100"      "  240"     " 28"
## [191,] " 4500"      "  347"     " 58"
## [192,] " 6000"      "   17"     " 77"
## [193,] "12000"      "  700"     "153"
## [194,] " 5850"      "  528"     " 88"
## [195,] " 1150"      "   50"     " 19"
## [196,] "  820"      "   16"     " 11"
## [197,] "12500"      " 1600"     "159"
## [198,] " 1200"      "  135"     " 16"
## [199,] " 4000"      "    1"     " 52"
## [200,] " 1990"      "   80"     " 26"
```

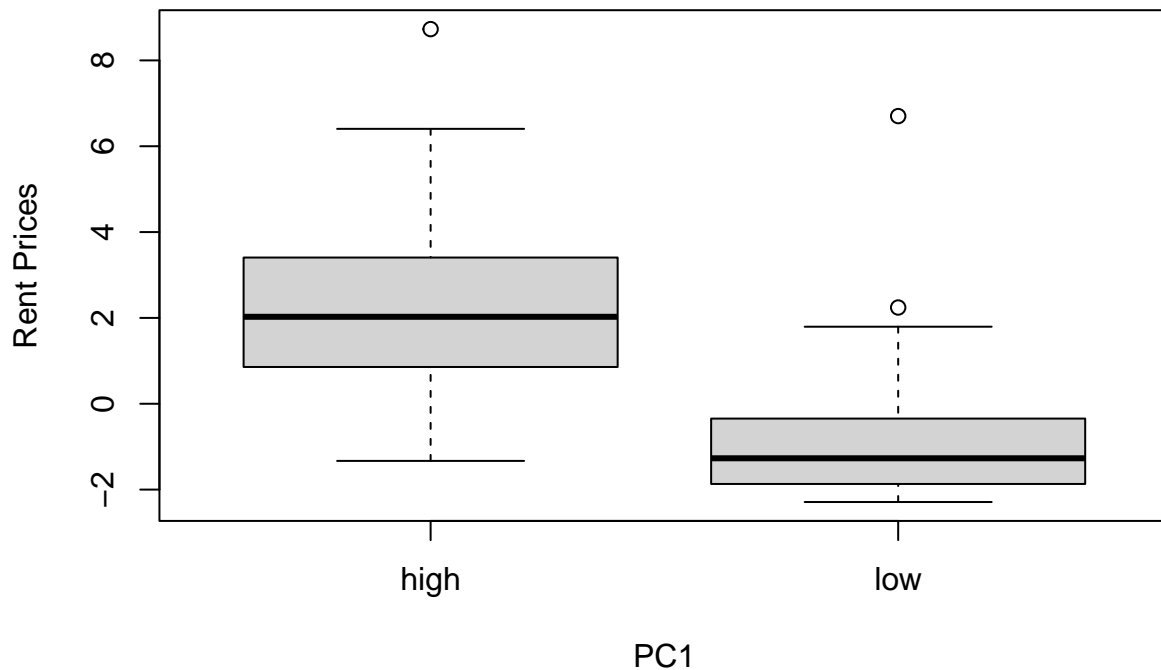
```
#drop(scale(new_rent,center=center, scale=scale))%*%rent_pca$rotation[,1])
predict(rent_pca)[,1]
```

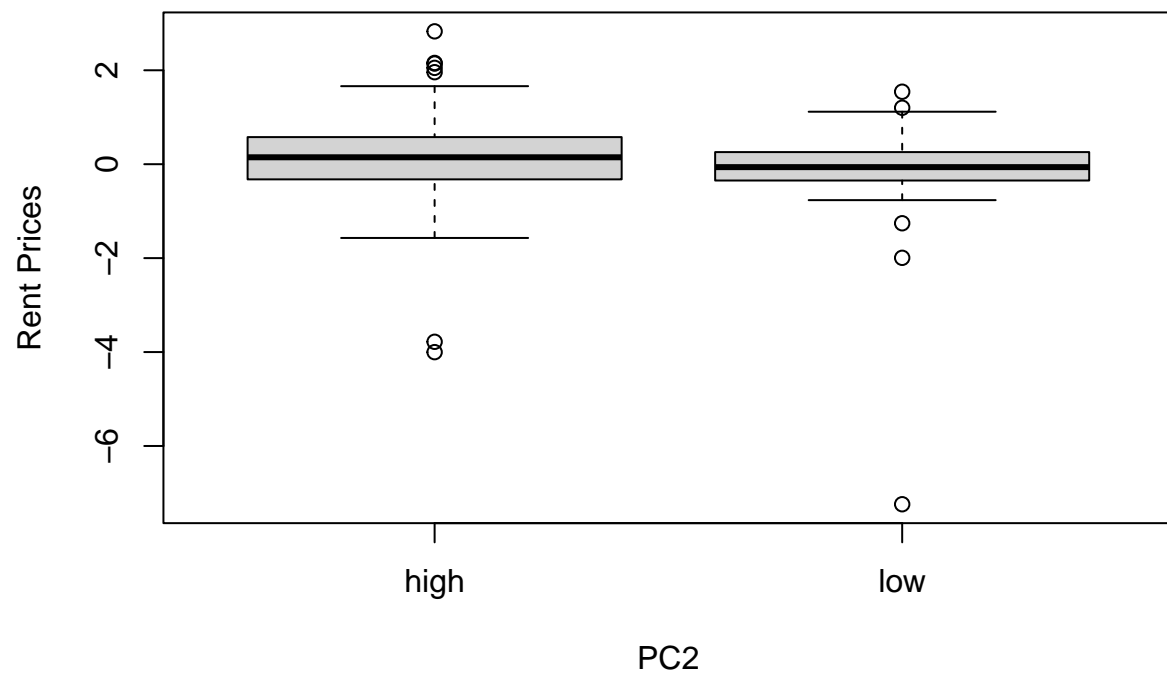
```
## [1] 1.56783825 -0.57639521 -1.50346548 -2.07514390 1.34530122 2.24975917
## [7] -2.23846999 -0.22912144 0.64873235 1.79476843 -0.95873902 2.13362330
## [13] -1.01660046 -2.07539668 -1.35466295 -0.76938889 -0.82924339 -1.67879982
## [19] 0.63656545 -2.03744269 -1.54280272 0.16777452 3.19501820 3.50284784
## [25] -0.32699280 -1.89708178 -2.05561487 -1.89149015 -0.87692112 1.85262375
## [31] -1.88522590 1.13964853 1.30826933 -1.22973070 2.24230187 1.94733013
## [37] 8.72831120 -0.61880179 -2.02535867 0.07631034 -2.08583746 2.94958392
## [43] 0.68623055 0.27919326 1.10879850 2.05071676 4.21959089 -1.27087429
## [49] 4.58113442 -2.02223866 -0.42267359 -0.90365748 -0.58003387 2.90035484
## [55] 6.70167586 -1.58788243 -2.08095556 -2.07987411 -1.23803192 -1.27346638
## [61] -2.06371912 0.37233762 0.16309312 -1.99481328 1.30865679 -0.39000716
## [67] 0.64357804 -1.72324081 -2.01219554 -2.11306394 -1.60270788 0.77543632
## [73] 0.52578996 -2.19000223 0.28960911 -1.76754748 -1.21365160 -1.80275583
## [79] -1.75325429 0.80748929 -0.80315118 -1.13014395 0.19883740 -0.85903456
## [85] -0.30037935 -1.47075763 -0.22483293 -2.28955542 -0.41144395 1.51846473
## [91] 0.89449869 -1.53674605 6.19640421 -1.86005657 0.33212522 0.80830540
## [97] 1.13265608 -0.91478591 -1.03005630 -0.22835744 4.55935064 -1.78927235
## [103] -1.46833723 -1.86780325 -1.58158590 3.40634597 2.81079495 -1.76194430
## [109] -1.95834325 -1.47171259 -2.01386835 5.76211882 -1.71655905 -1.36509520
```

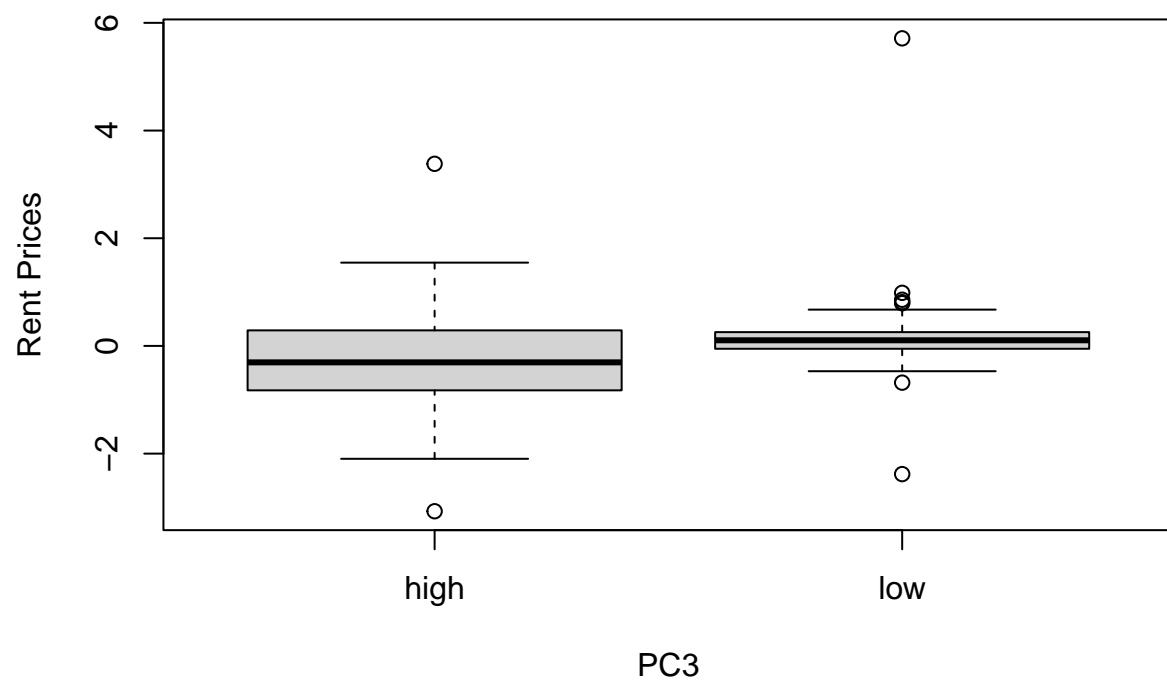
```
## [115] -0.65189695  0.13935881  1.93908906 -0.36387251  3.06217521  0.21859561
## [121] -1.76243407 -0.43837393 -1.02785008 -2.09994961 -2.05366442 -2.13691077
## [127] -1.79886574 -0.73346935 -0.52651398 -1.76290055 -2.19205769 -1.60079651
## [133] -0.94453039 -0.57639521  4.53213015 -1.20831130 -1.99402495 -1.74622046
## [139] -0.65198241  2.05137269  0.87380363 -2.12397932  0.96227605 -1.01666875
## [145]  1.57023222 -0.04858875  3.68932809 -1.91380925 -0.56676741 -2.21839448
## [151] -1.28894407 -0.15718144 -1.38189083 -0.47778490 -2.14612555  2.62293011
## [157] -0.31225768  3.76474618  0.85651631  2.11184808 -2.15956186 -1.34879407
## [163] -0.60762535  2.02586190 -1.05921271 -1.03323781 -0.10756112  3.97320720
## [169]  6.40500435 -1.56351179  4.70912592  0.91652794  3.01780491 -0.14094557
## [175] -1.95900629  1.89765204 -2.16356341 -2.04455537 -0.08406563 -1.87051599
## [181] -0.15840553  2.57672105  1.93043286 -0.21811819 -1.66127191 -1.84040471
## [187]  4.40504874  0.27881638 -0.16493909 -0.50611546  0.65734764  0.96346403
## [193]  3.16873531  1.58293083 -1.43496833 -1.41999697  5.01064356 -0.49933014
## [199] -1.33180077 -1.21380272
```

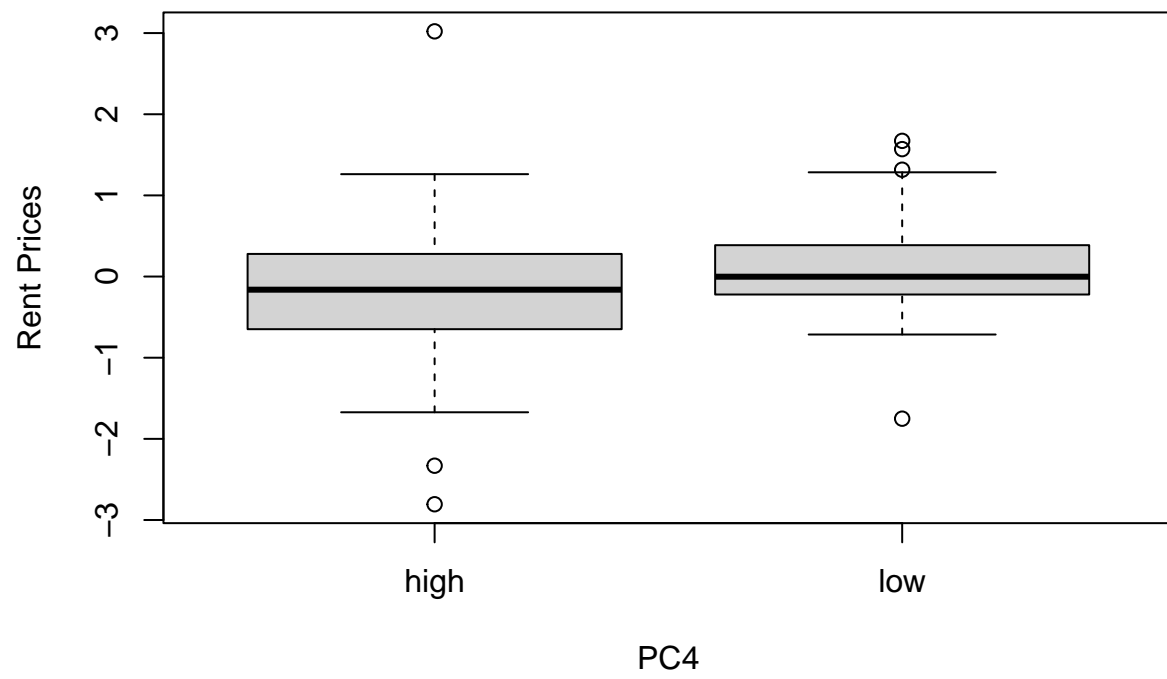
#The aboved two gives us the same thing. predict is a good function to know.

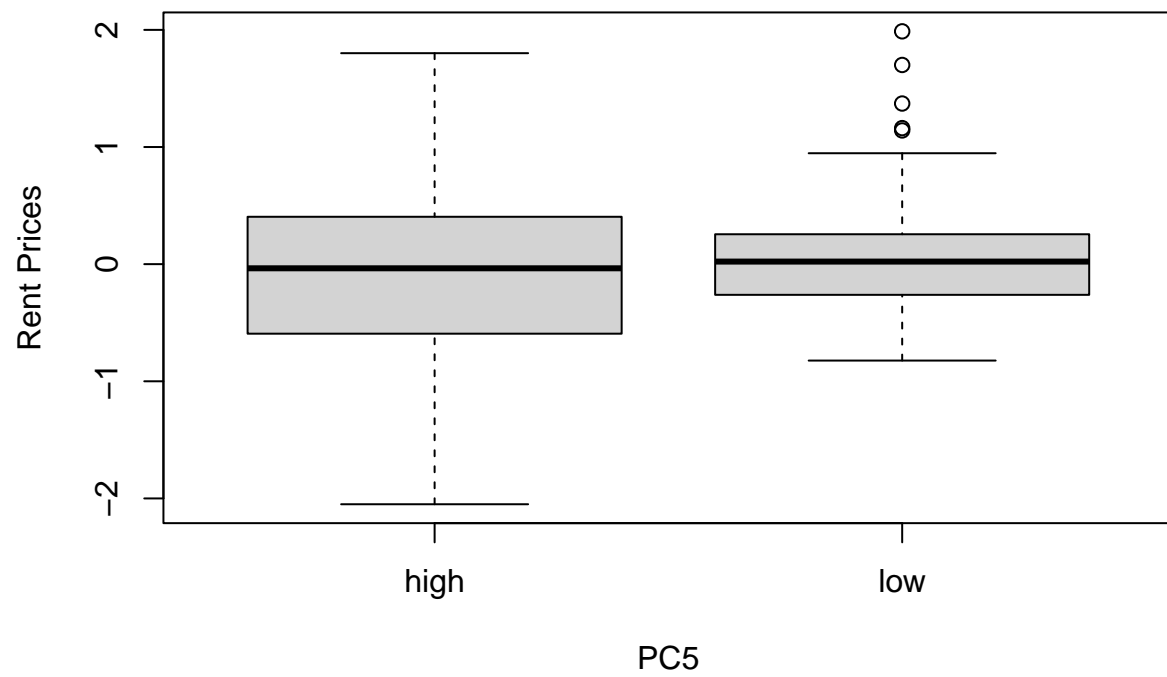
```
house_data$rent <- as.factor(house_data$rent)
out <- sapply(1:5, function(i){plot(house_data$rent,rent_pca$x[,i],xlab=paste("PC",i,sep=""),ylab="Rent
```



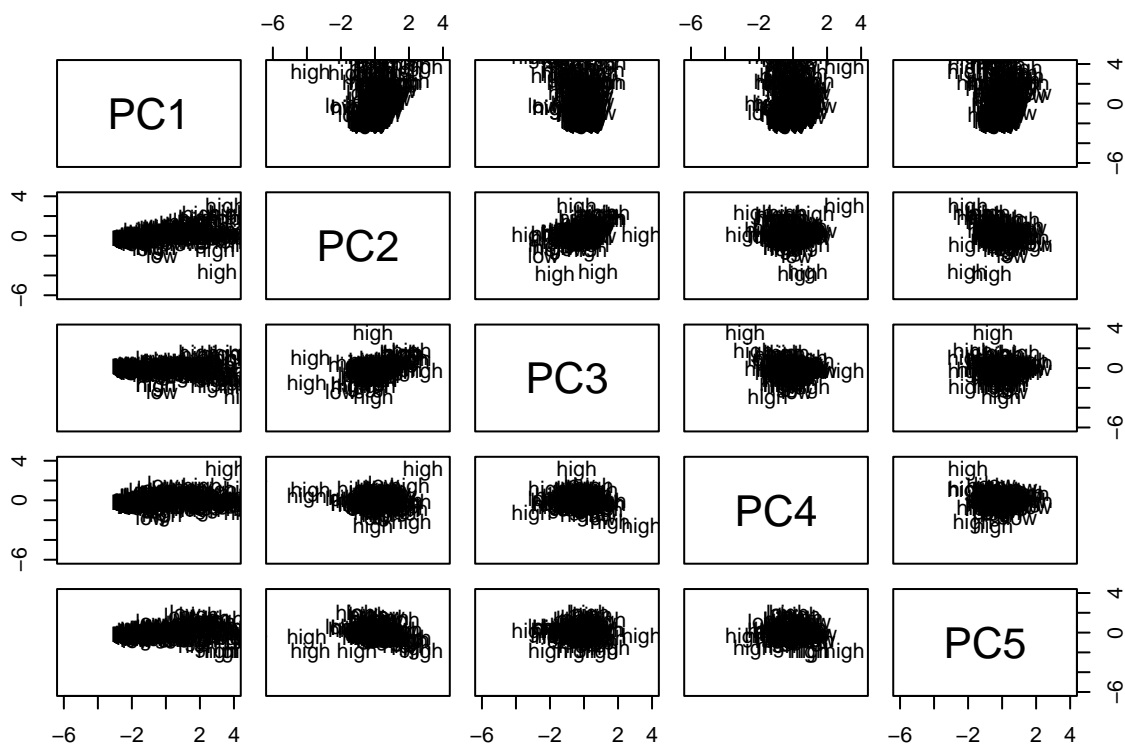








```
pairs(rent_pca$x[,1:5], ylim = c(-6,4),xlim = c(-6,4),panel=function(x,y,...){text(x,y,house_data$rent)})
```



Visualization

Question 3 Perform some visualization using PCs

```
library(factoextra)
```

```
## Loading required package: ggplot2
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(FactoMineR)
library(ggfortify)
library(psych)
```

```
##
```

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

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

```
##
```

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

```
## The following object is masked from 'package:car':
##
##   logit
```

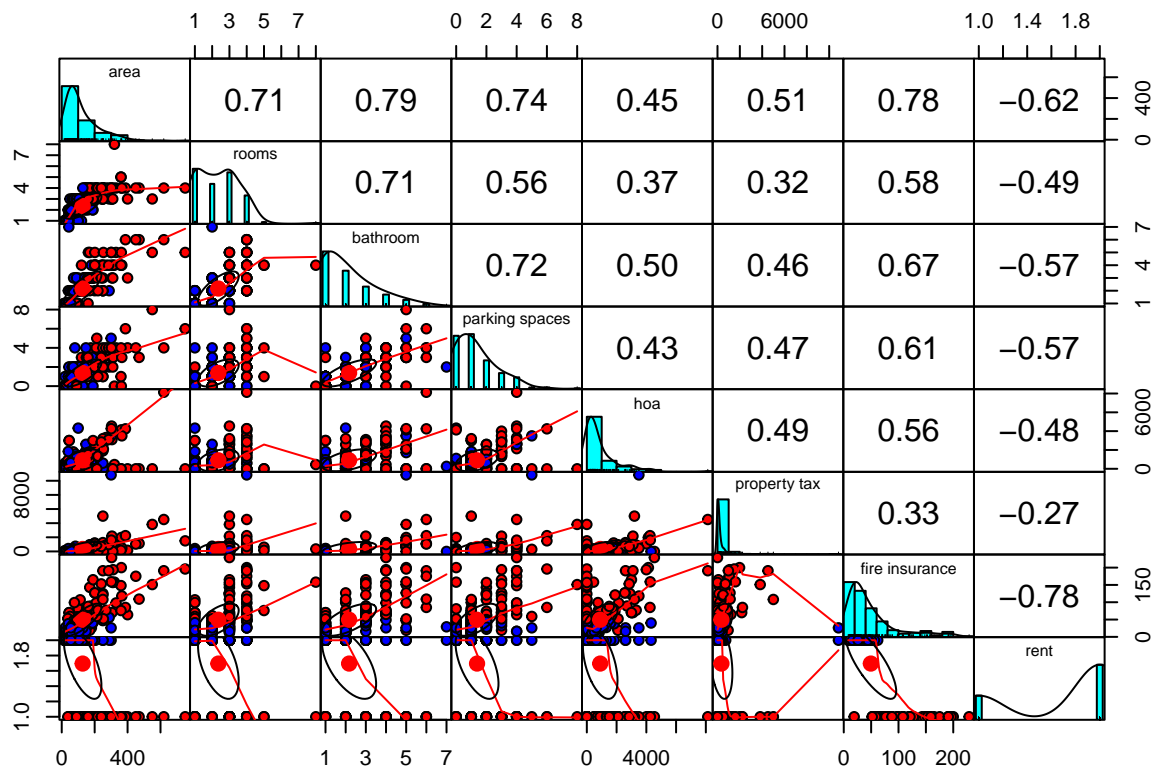
```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

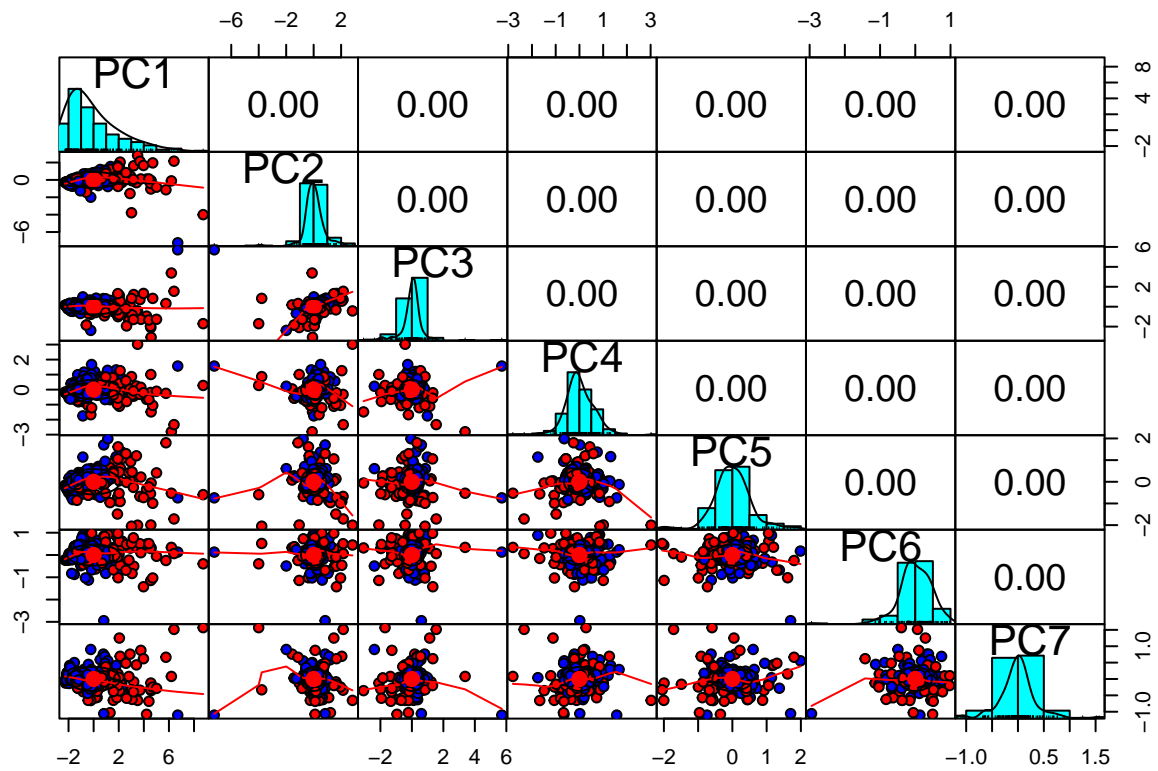
```
library(devtools)
```

```
## Loading required package: usethis
```

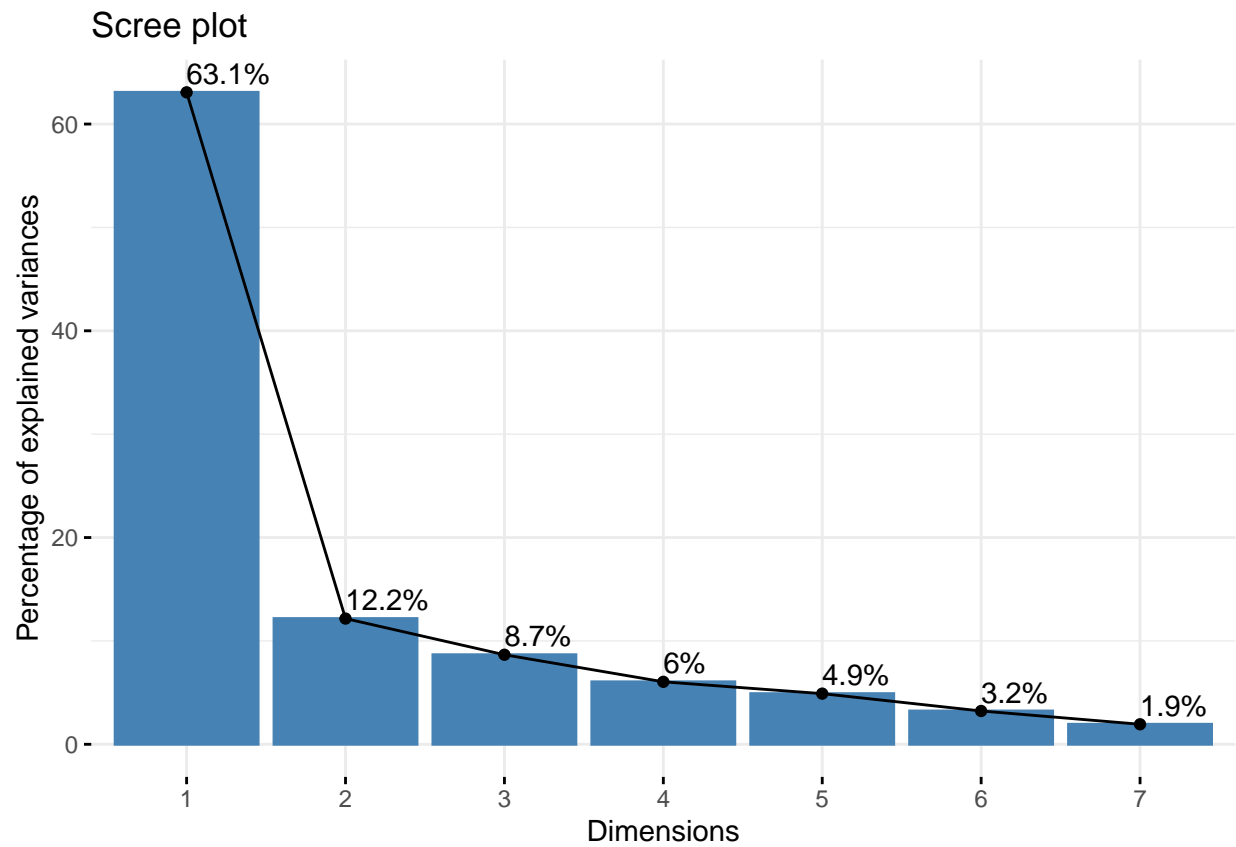
```
# Correlation
pairs.panels(house_data[, -c(5,6,8)],
             gap = 0,
             bg = c("red", "blue")[house_data$rent],
             pch=21)
```



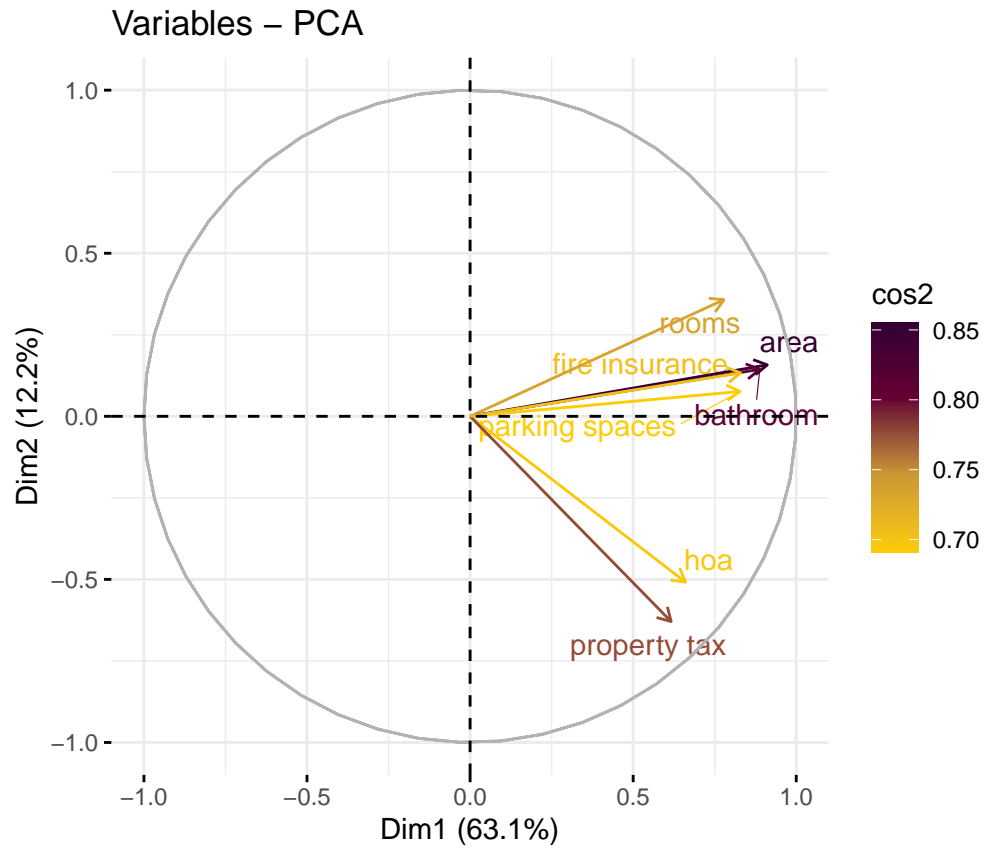
```
pairs.panels(rent_pca$x,
             gap=0,
             bg = c("red", "blue")[house_data$rent],
             pch=21)
```



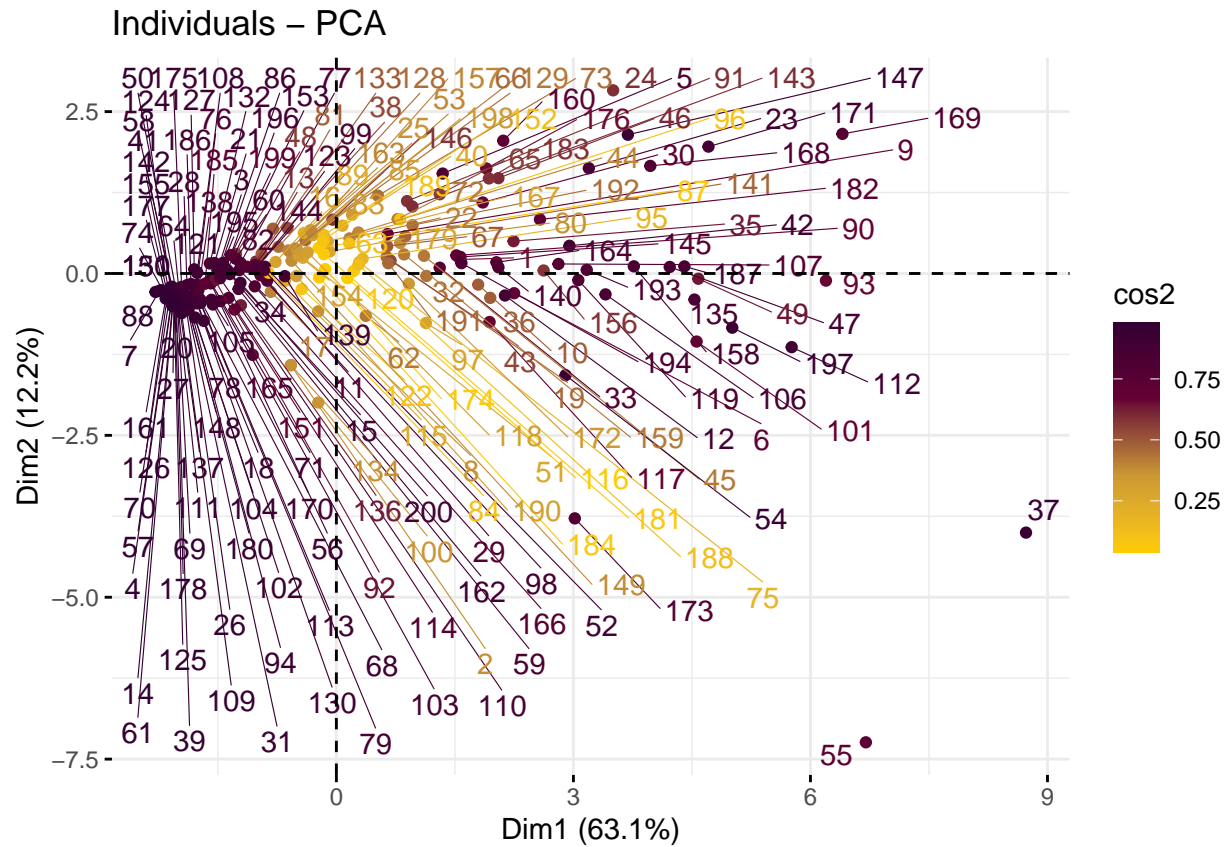
```
fviz_eig(rent_pca, addlabels = TRUE)
```



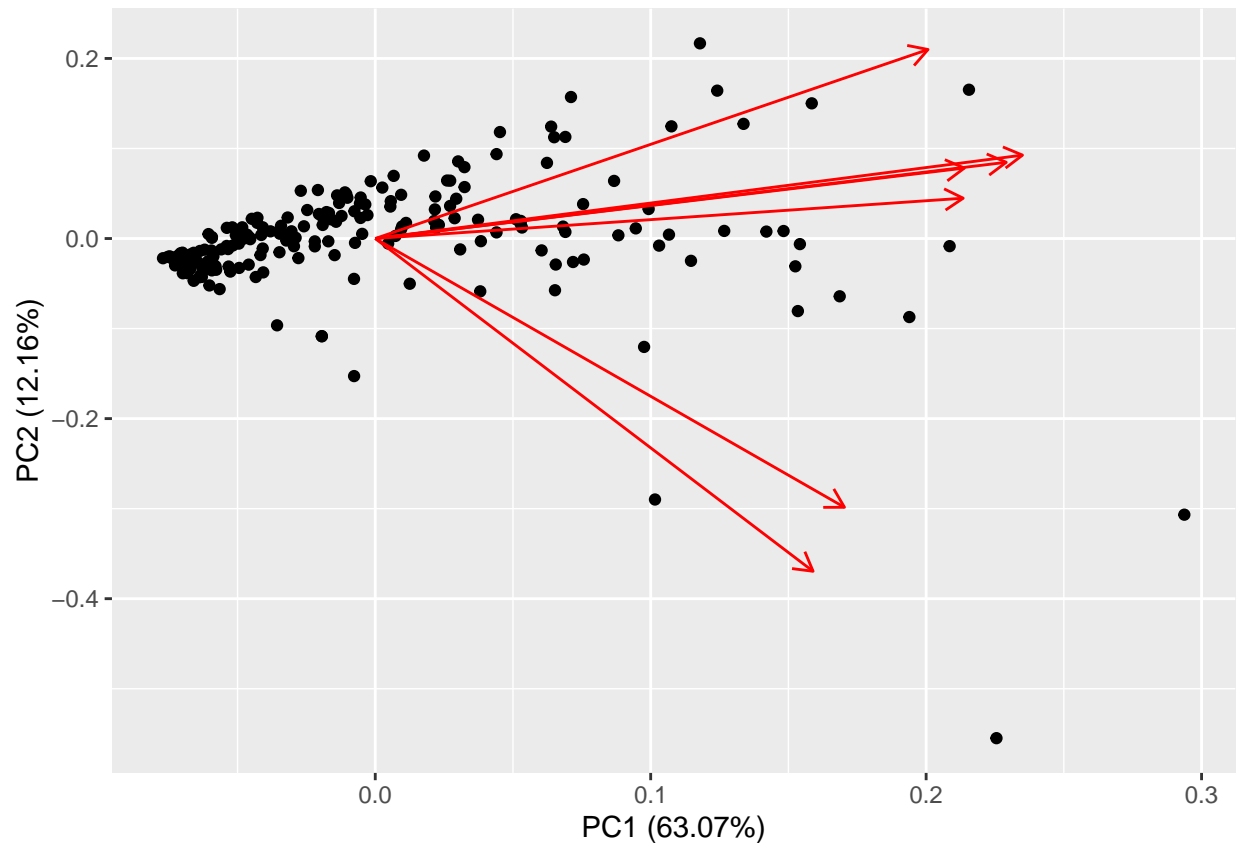
```
fviz_pca_var(rent_pca,col.var = "cos2",  
             gradient.cols = c("#FFCC00", "#CC9933", "#660033", "#330033"),  
             repel = TRUE)
```

```
fviz_pca_ind(rent_pca, col.ind = "cos2",  
             gradient.cols = c("#FFCC00", "#CC9933", "#660033", "#330033"),  
             repel = TRUE)
```



```
biplot(rent_pca)
```

```
# Different PCA Method.
```

```
res.pca <- PCA(house_data[, -c(5,6,8,11)], graph = FALSE)
print(res.pca)
```

```
## **Results for the Principal Component Analysis (PCA)**
## The analysis was performed on 200 individuals, described by 7 variables
## *The results are available in the following objects:
##
##   name          description
## 1  "$eig"        "eigenvalues"
## 2  "$var"        "results for the variables"
## 3  "$var$coord"  "coord. for the variables"
## 4  "$var$cor"    "correlations variables - dimensions"
## 5  "$var$cos2"   "cos2 for the variables"
## 6  "$var$contrib" "contributions of the variables"
## 7  "$ind"        "results for the individuals"
## 8  "$ind$coord"  "coord. for the individuals"
## 9  "$ind$cos2"   "cos2 for the individuals"
## 10 "$ind$contrib" "contributions of the individuals"
## 11 "$call"       "summary statistics"
## 12 "$call$centre" "mean of the variables"
## 13 "$call$ecart.type" "standard error of the variables"
## 14 "$call$row.w"  "weights for the individuals"
## 15 "$call$col.w"  "weights for the variables"
```

```
# Visualize and Interpret PCA using these functions
```

```
#get_eigenvalue(res.pca): Extract the eigenvalues/variances of principal components
```

```
#fviz_eig(res.pca): Visualize the eigenvalues
```

```
#get_pca_ind(res.pca), get_pca_var(res.pca): Extract the results for individuals and variables, respect
```

```
#fviz_pca_ind(res.pca), fviz_pca_var(res.pca): Visualize the results individuals and variables, respect
```

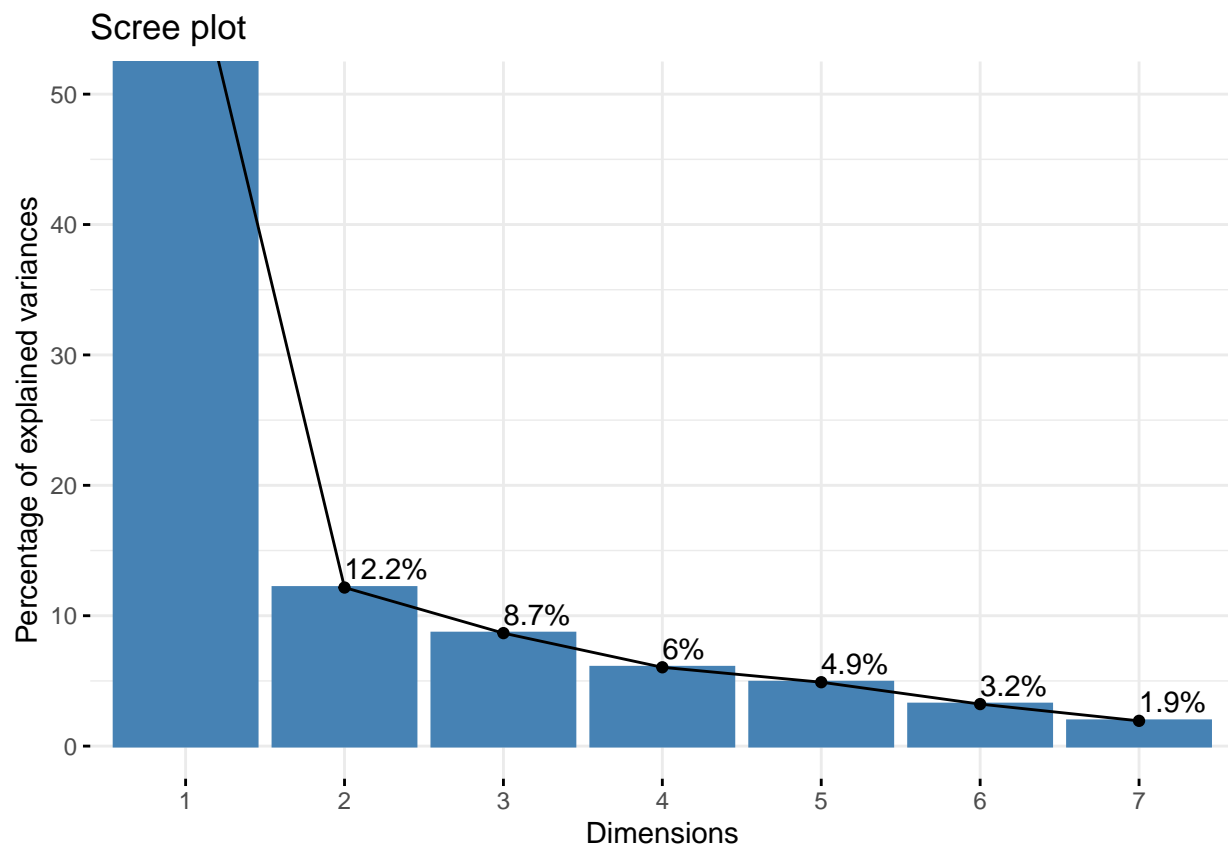
```
#fviz_pca_biplot(res.pca): Make a biplot of individuals and variables.
```

```
eig.val <- get_eigenvalue(res.pca)
```

```
eig.val
```

##	eigenvalue	variance.percent	cumulative.variance.percent
## Dim.1	4.4147952	63.068502	63.06850
## Dim.2	0.8513135	12.161621	75.23012
## Dim.3	0.6064989	8.664270	83.89439
## Dim.4	0.4231021	6.044316	89.93871
## Dim.5	0.3430459	4.900656	94.83937
## Dim.6	0.2255804	3.222577	98.06194
## Dim.7	0.1356640	1.938057	100.00000

```
fviz_eig(res.pca, addlabels = TRUE, ylim = c(0, 50))
```



```
var <- get_pca_var(res.pca)
```

```
#var$coord: coordinates of variables to create a scatter plot
```

*#var\$cos2: represents the quality of representation for variables on the factor map. It's calculated as
 #var\$contrib: contains the contributions (in percentage) of the variables to the principal components.
 #The contribution of a variable (var) to a given principal component is (in percentage) : (var.cos2 * 100)
 var*

```
## Principal Component Analysis Results for variables
## =====
##   Name      Description
## 1 "$coord"   "Coordinates for the variables"
## 2 "$cor"     "Correlations between variables and dimensions"
## 3 "$cos2"    "Cos2 for the variables"
## 4 "$contrib" "contributions of the variables"
```

Coordinates
`head(var$coord)`

	Dim.1	Dim.2	Dim.3	Dim.4	Dim.5
## area	0.9112576	-0.15770124	-0.09338617	-0.09277482	-0.20641197
## rooms	0.7782706	-0.35745952	-0.05286832	0.47538513	0.01405257
## bathroom	0.8883542	-0.14418343	-0.05145417	0.06314732	0.16751380
## parking spaces	0.8277606	-0.07628277	-0.21666018	-0.33861446	0.33209675
## hoa	0.6606244	0.50827186	0.50036736	0.11371340	0.18330524
## property tax	0.6164326	0.62925244	-0.42367527	0.08082836	-0.17888639

Cos2: quality on the factore map
`head(var$cos2)`

	Dim.1	Dim.2	Dim.3	Dim.4	Dim.5
## area	0.8303904	0.024869680	0.008720977	0.008607167	0.0426058993
## rooms	0.6057051	0.127777310	0.002795060	0.225991026	0.0001974747
## bathroom	0.7891731	0.020788861	0.002647531	0.003987584	0.0280608726
## parking spaces	0.6851876	0.005819061	0.046941632	0.114659749	0.1102882493
## hoa	0.4364246	0.258340282	0.250367497	0.012930736	0.0336008116
## property tax	0.3799891	0.395958629	0.179500738	0.006533224	0.0320003400

Contributions to the principal components
`head(var$contrib)`

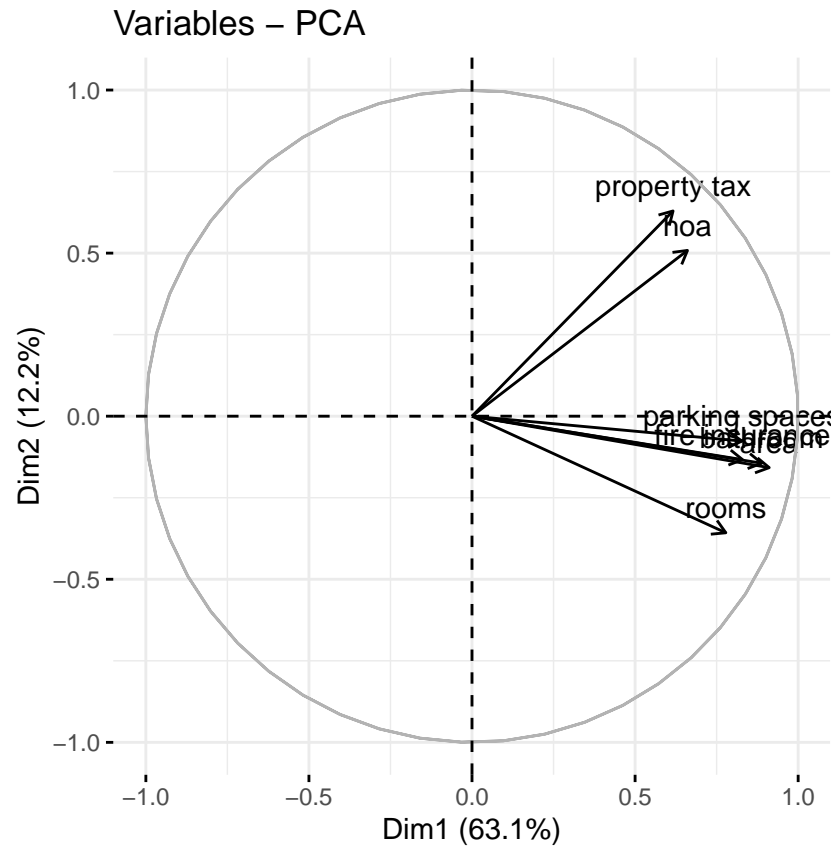
	Dim.1	Dim.2	Dim.3	Dim.4	Dim.5
## area	18.809263	2.9213304	1.4379213	2.0343000	12.4198829
## rooms	13.719892	15.0094309	0.4608515	53.4128775	0.0575651
## bathroom	17.875645	2.4419748	0.4365270	0.9424637	8.1799177
## parking spaces	15.520259	0.6835392	7.7397719	27.0997802	32.1497060
## hoa	9.885501	30.3460812	41.2807826	3.0561737	9.7948442
## property tax	8.607174	46.5114948	29.5962176	1.5441245	9.3282968

*#The plot Below is also known as variable correlation plots. It shows the relationships between all variables.
 #Positively correlated variables are grouped together.
 #Negatively correlated variables are positioned on opposite sides of the plot origin (opposed quadrants).
 #The distance between variables and the origin measures the quality of the variables on the factor map.*

```
#Variables that are away from the origin are well represented on the factor map.
```

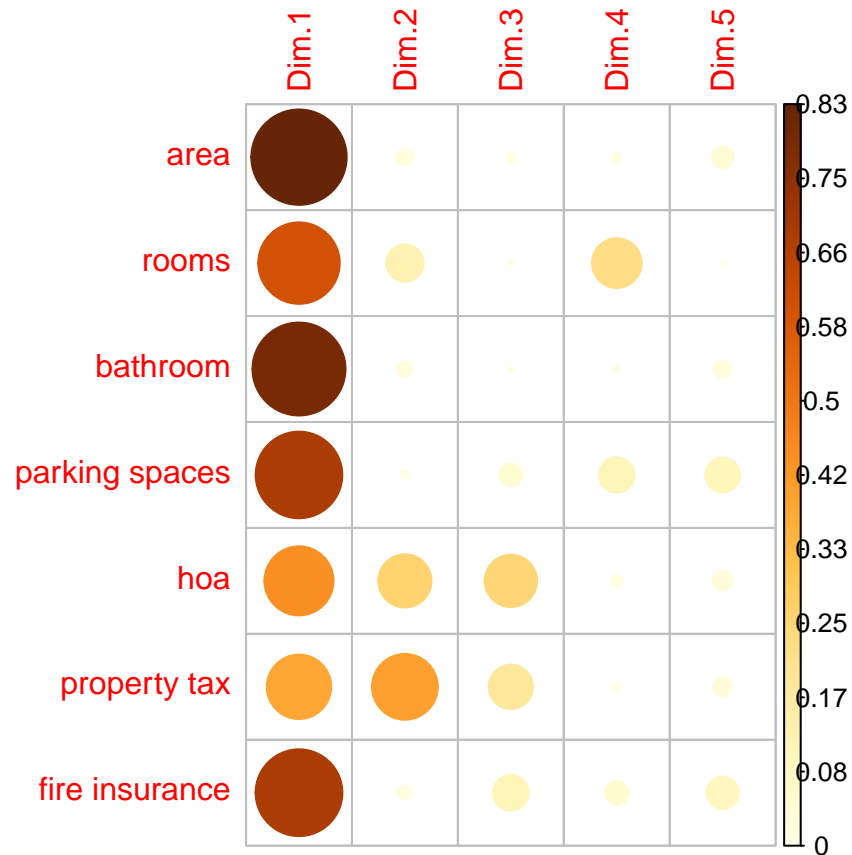
```
# Correlation circle
```

```
fviz_pca_var(res.pca, col.var = "black")
```



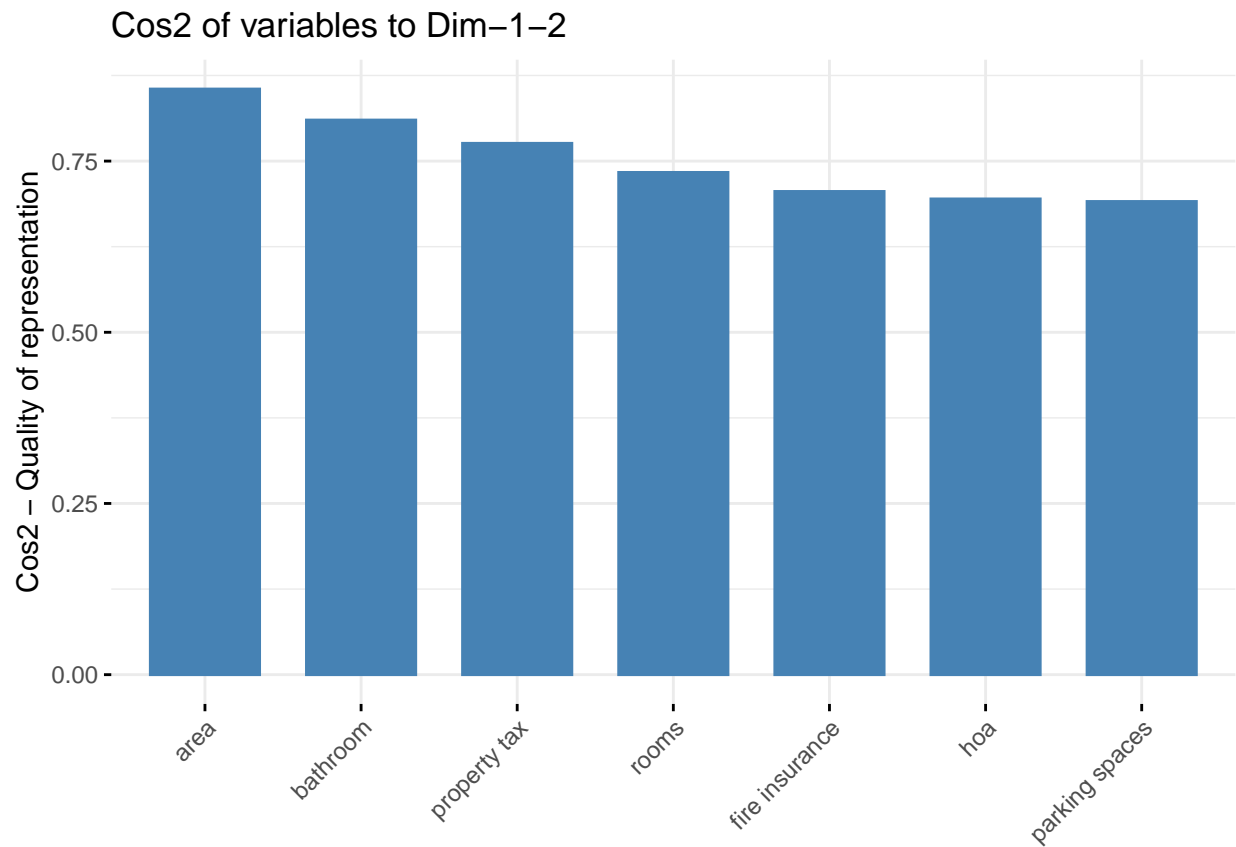
```
# Quality of representation
```

```
corrplot(var$cos2, is.corr=FALSE)
```

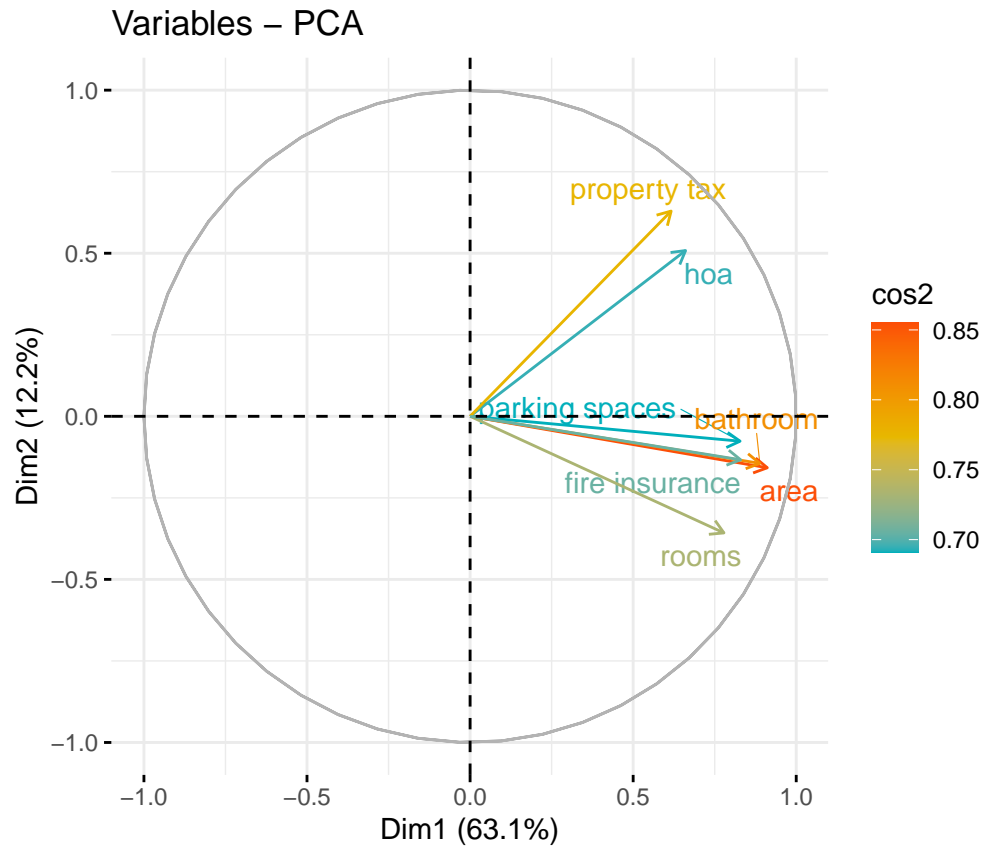


Total cos2 of variables on Dim.1 and Dim.2
#A high cos2 indicates a good representation of the variable on the principal component.
#In this case the variable is positioned close to the circumference of the correlation circle.
#A low cos2 indicates that the variable is not perfectly represented by the PCs.
#In this case the variable is close to the center of the circle.

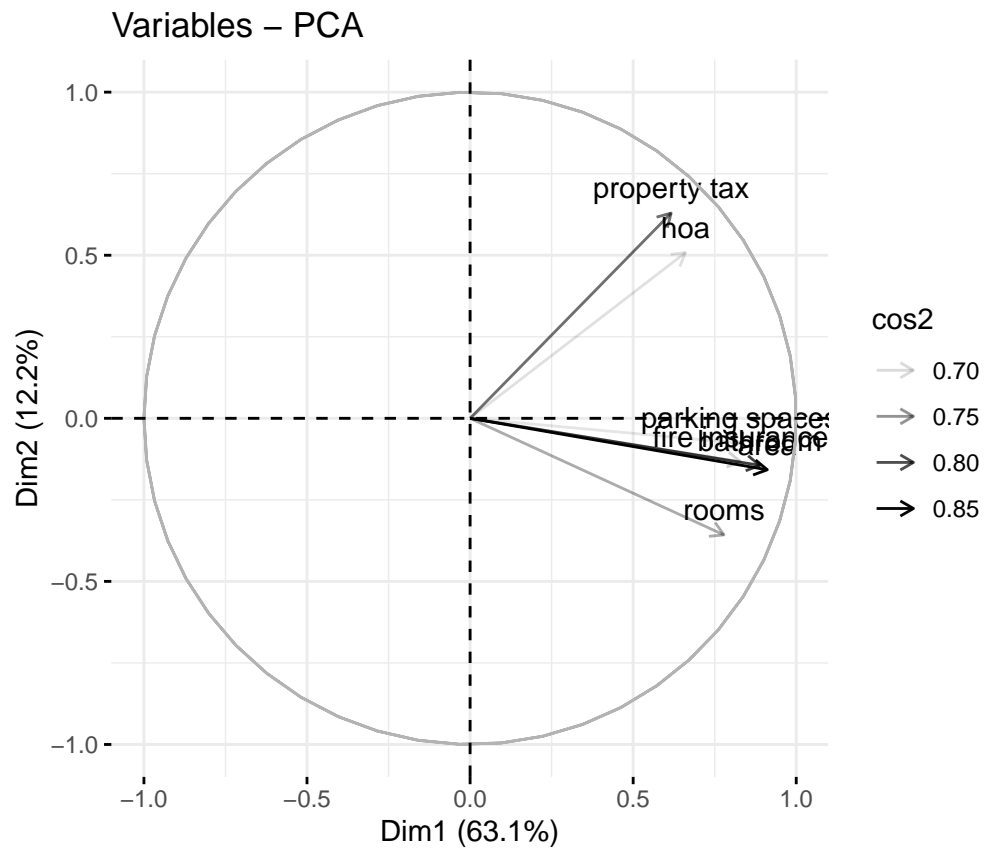
```
fviz_cos2(res.pca, choice = "var", axes = 1:2)
```

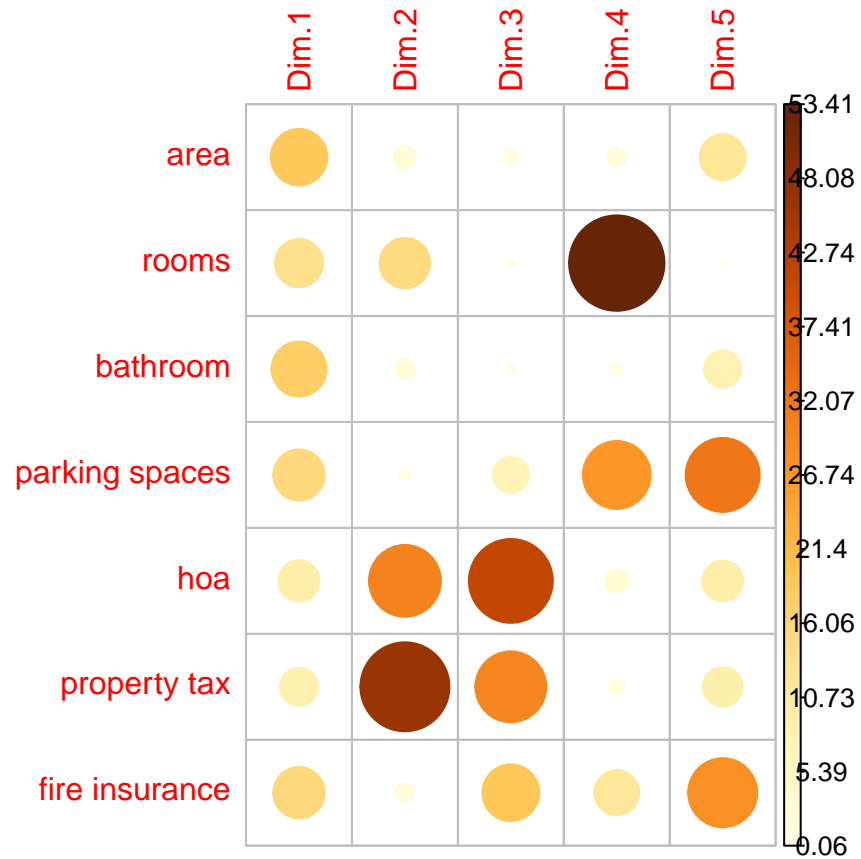
```
fviz_pca_var(res.pca, col.var = "cos2",  
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),  
             repel = TRUE # Avoid text overlapping  
             )
```



```
# Change the transparency by cos2 values  
fviz_pca_var(res.pca, alpha.var = "cos2")
```

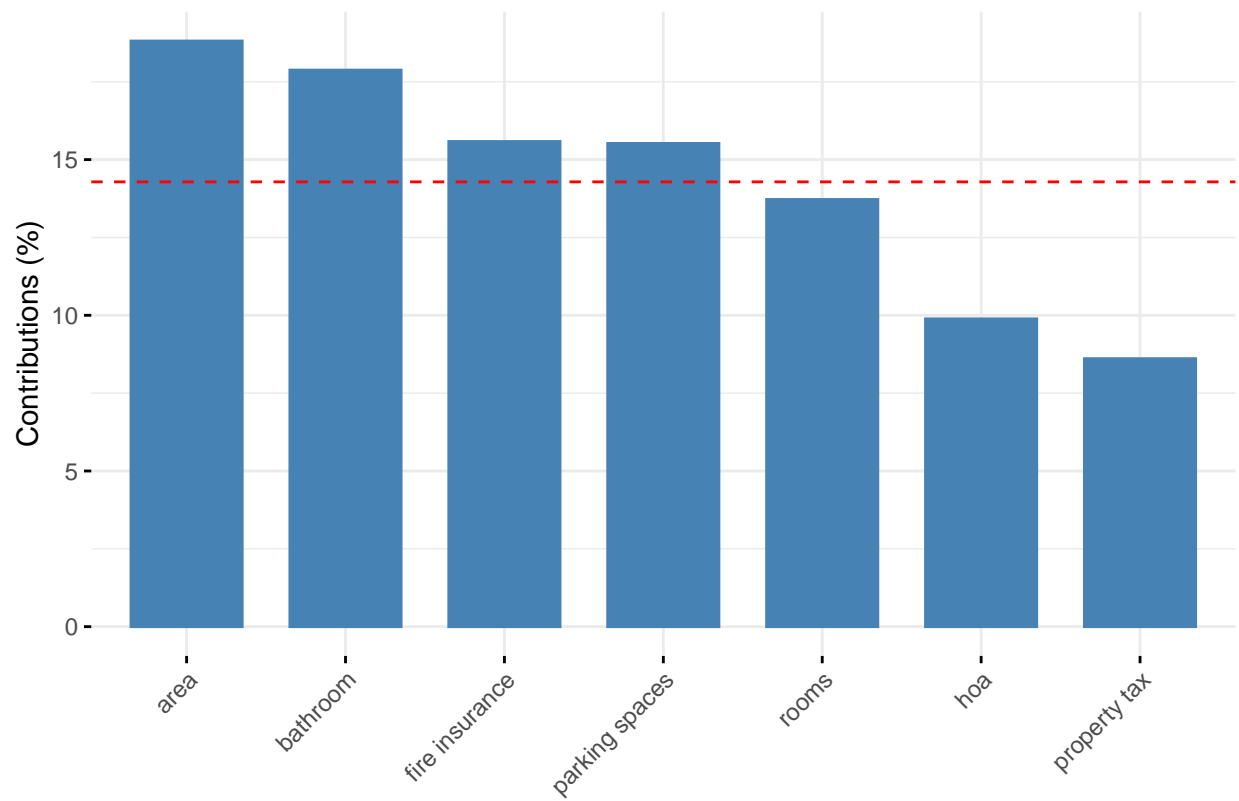


```
corrplot(var$contrib, is.corr=FALSE)
```

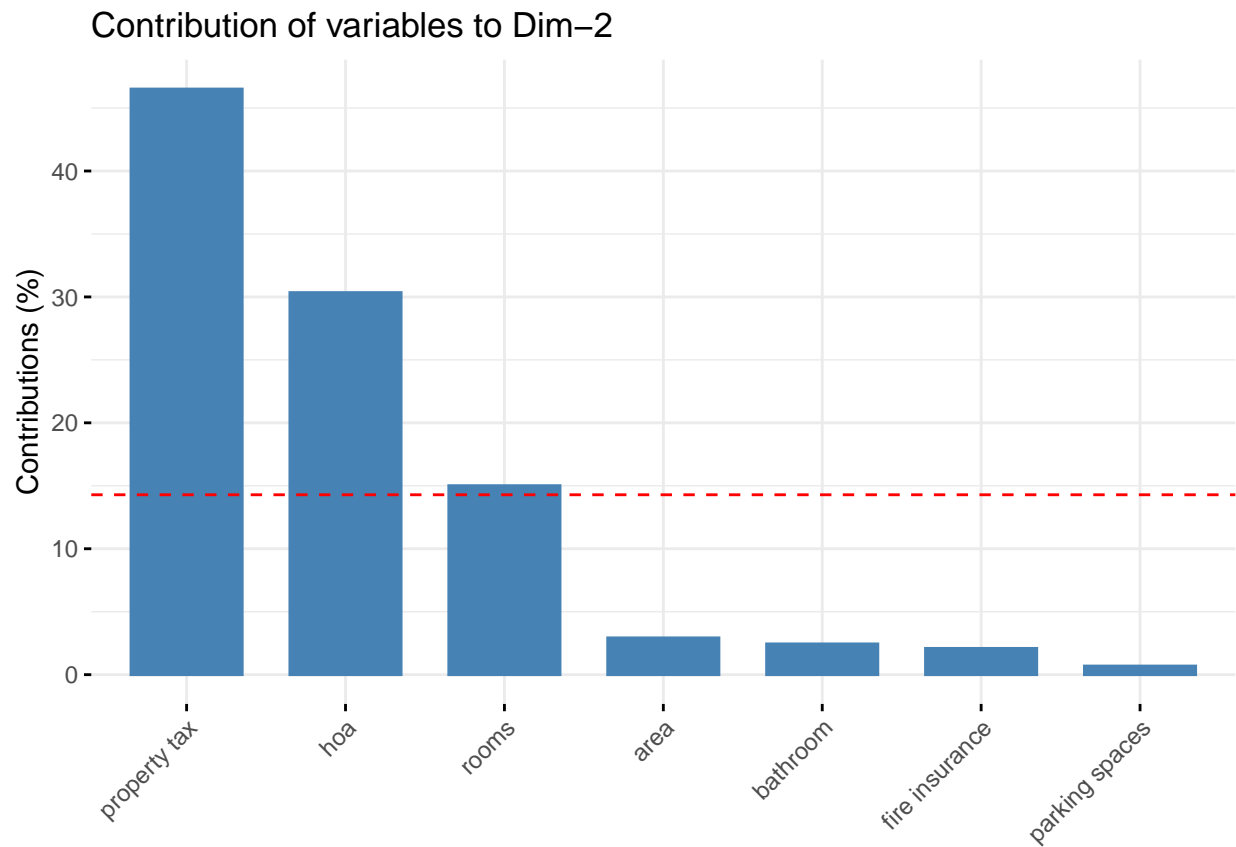


```
# Contributions of variables to PC1
fviz_contrib(res.pca, choice = "var", axes = 1, top = 10)
```

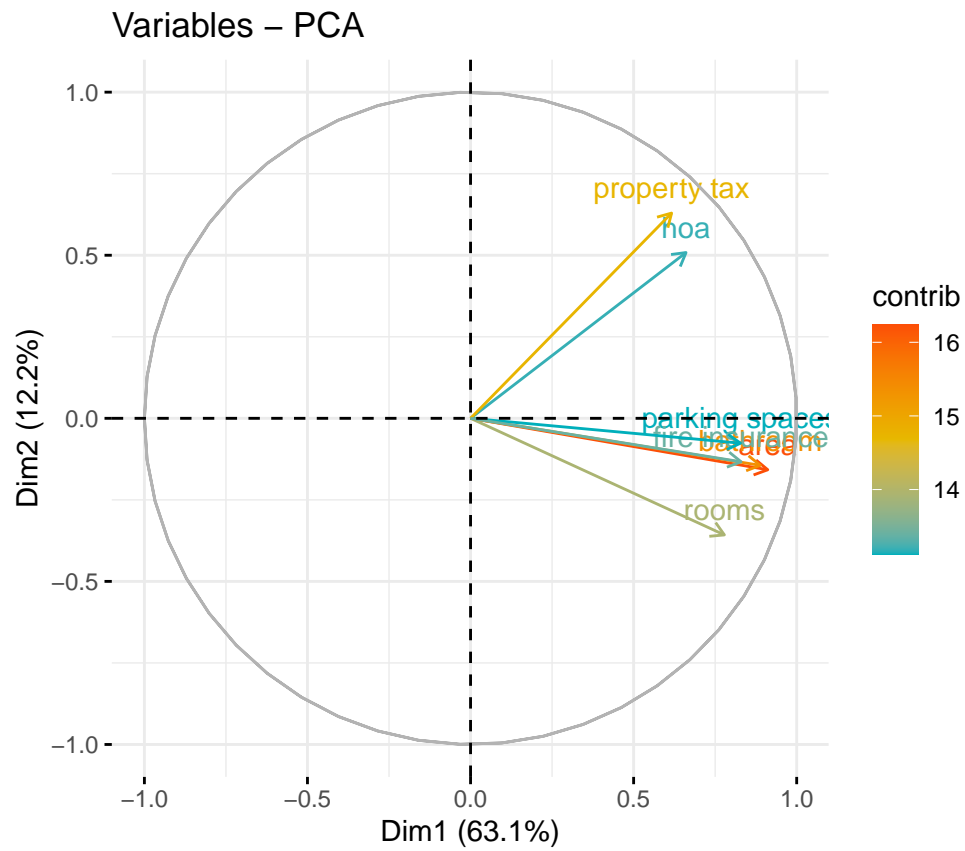
Contribution of variables to Dim-1



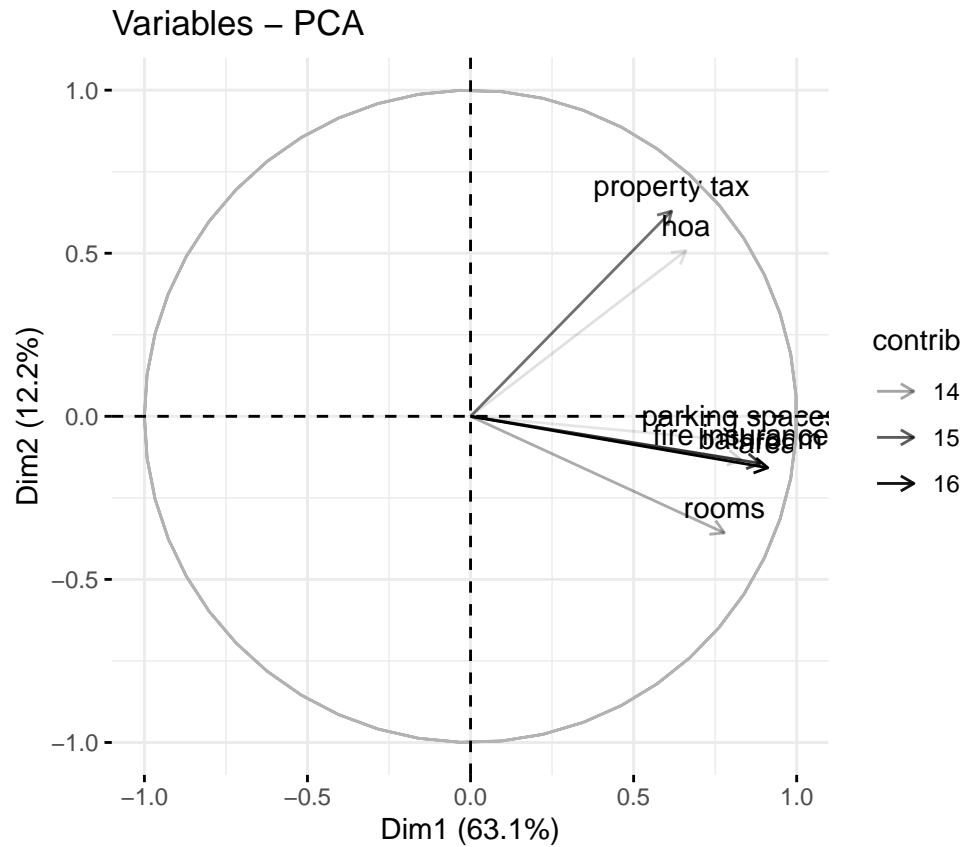
```
# Contributions of variables to PC2  
fviz_contrib(res.pca, choice = "var", axes = 2, top = 10)
```



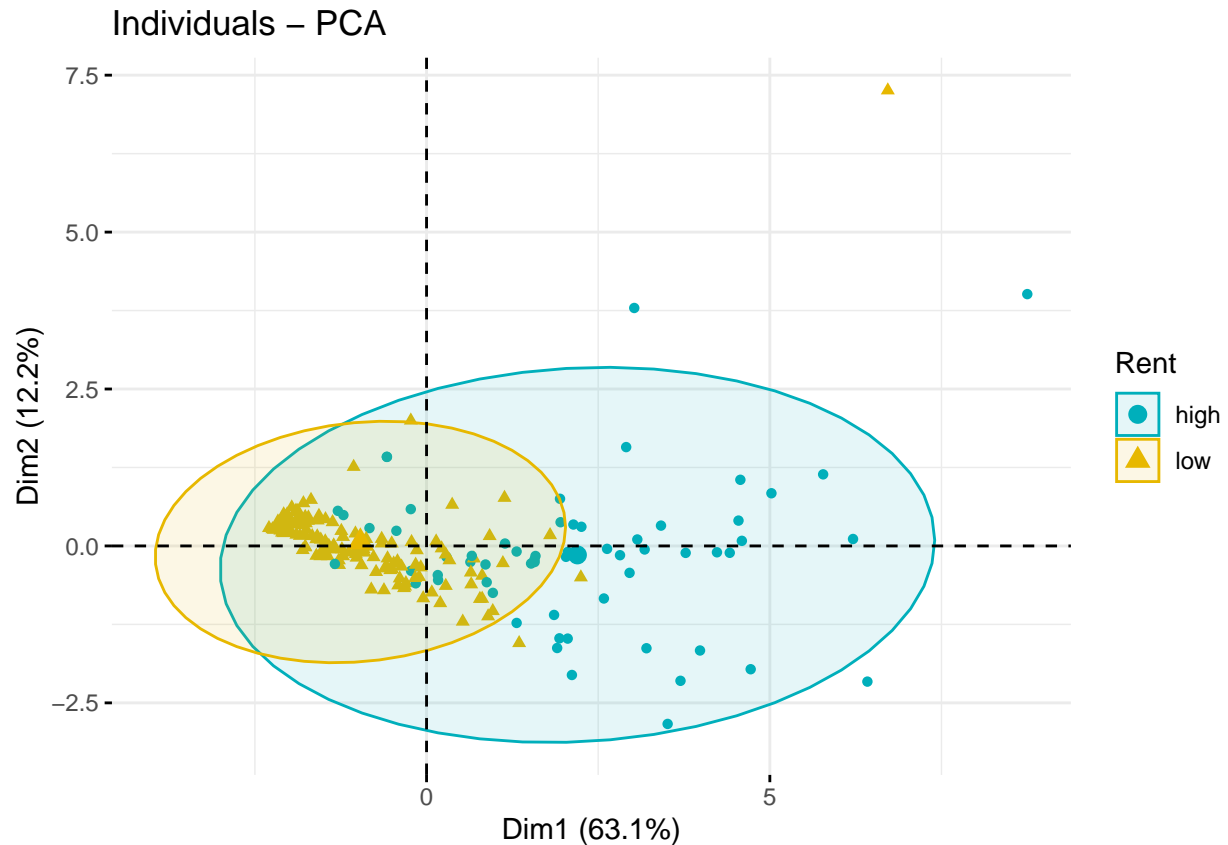
```
fviz_pca_var(res.pca, col.var = "contrib",  
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07")  
             )
```



```
fviz_pca_var(res.pca, alpha.var = "contrib")
```



```
fviz_pca_ind(res.pca,
  geom.ind = "point", # show points only (nbut not "text")
  col.ind = house_data$rent, # color by groups
  palette = c("#00AFBB", "#E7B800", "#FC4E07"),
  addEllipses = TRUE, # Concentration ellipses
  legend.title = "Rent"
)
```

```
# Description of PC
```

```
res.desc <- dimdesc(res.pca, axes = c(1,2,3,4,5), proba = 0.05)
```

```
# Description of dimension 1
```

```
res.desc$Dim.1
```

```
##
```

```
## Link between the variable and the continuous variables (R-square)
```

```
## =====
```

	correlation	p.value
area	0.9112576	3.225242e-78
bathroom	0.8883542	7.457100e-69
fire insurance	0.8294125	5.820123e-52
parking spaces	0.8277606	1.384481e-51
rooms	0.7782706	7.027759e-42
hoa	0.6606244	1.881880e-26
property tax	0.6164326	2.554065e-22

```
res.desc$Dim.2
```

```
##
```

```
## Link between the variable and the continuous variables (R-square)
```

```
## =====
```

	correlation	p.value
property tax	0.6292524	1.890530e-23

```
## hoa          0.5082719 1.553104e-14
## bathroom    -0.1441834 4.165641e-02
## area        -0.1577012 2.573237e-02
## rooms       -0.3574595 2.033302e-07
```

```
res.desc$Dim.3
```

```
##
## Link between the variable and the continuous variables (R-square)
## =====
##          correlation      p.value
## hoa          0.5003674 4.544495e-14
## fire insurance 0.3398904 8.477789e-07
## parking spaces -0.2166602 2.059478e-03
## property tax  -0.4236753 4.077071e-10
```

```
res.desc$Dim.4
```

```
##
## Link between the variable and the continuous variables (R-square)
## =====
##          correlation      p.value
## rooms          0.4753851 1.134282e-12
## fire insurance -0.2244831 1.395134e-03
## parking spaces -0.3386145 9.372570e-07
```

```
res.desc$Dim.5
```

```
##
## Link between the variable and the continuous variables (R-square)
## =====
##          correlation      p.value
## parking spaces 0.3320967 1.553949e-06
## hoa           0.1833052 9.373282e-03
## bathroom      0.1675138 1.774331e-02
## property tax  -0.1788864 1.126222e-02
## area          -0.2064120 3.362943e-03
## fire insurance -0.3103099 7.755482e-06
```

```
# Graph of Individuals
ind <- get_pca_ind(res.pca)
ind
```

```
## Principal Component Analysis Results for individuals
## =====
##   Name      Description
## 1 "$coord"   "Coordinates for the individuals"
## 2 "$cos2"    "Cos2 for the individuals"
## 3 "$contrib" "contributions of the individuals"
```

```
## Principal Component Analysis Results for individuals
## =====
##   Name      Description
## 1 "$coord"   "Coordinates for the individuals"
## 2 "$cos2"    "Cos2 for the individuals"
## 3 "$contrib" "contributions of the individuals"
##To get access to the different components, use this:

# Coordinates of individuals
head(ind$coord)
```

```
##          Dim.1      Dim.2      Dim.3      Dim.4      Dim.5
## 1  1.5717726 -0.21664876 -0.10090505 -0.12196643  0.8848836
## 2 -0.5778416  1.41917410  1.84967872 -0.56948043  0.2654308
## 3 -1.5072383 -0.04751288 -0.31545227  0.07614962  0.2096122
## 4 -2.0803513  0.35429625  0.04890106 -0.21408346 -0.2472104
## 5  1.3486771 -1.54706351 -0.62774604  0.56784111  0.1271920
## 6  2.2554047  0.30526093  0.63757865 -0.57095206  1.0688021
```

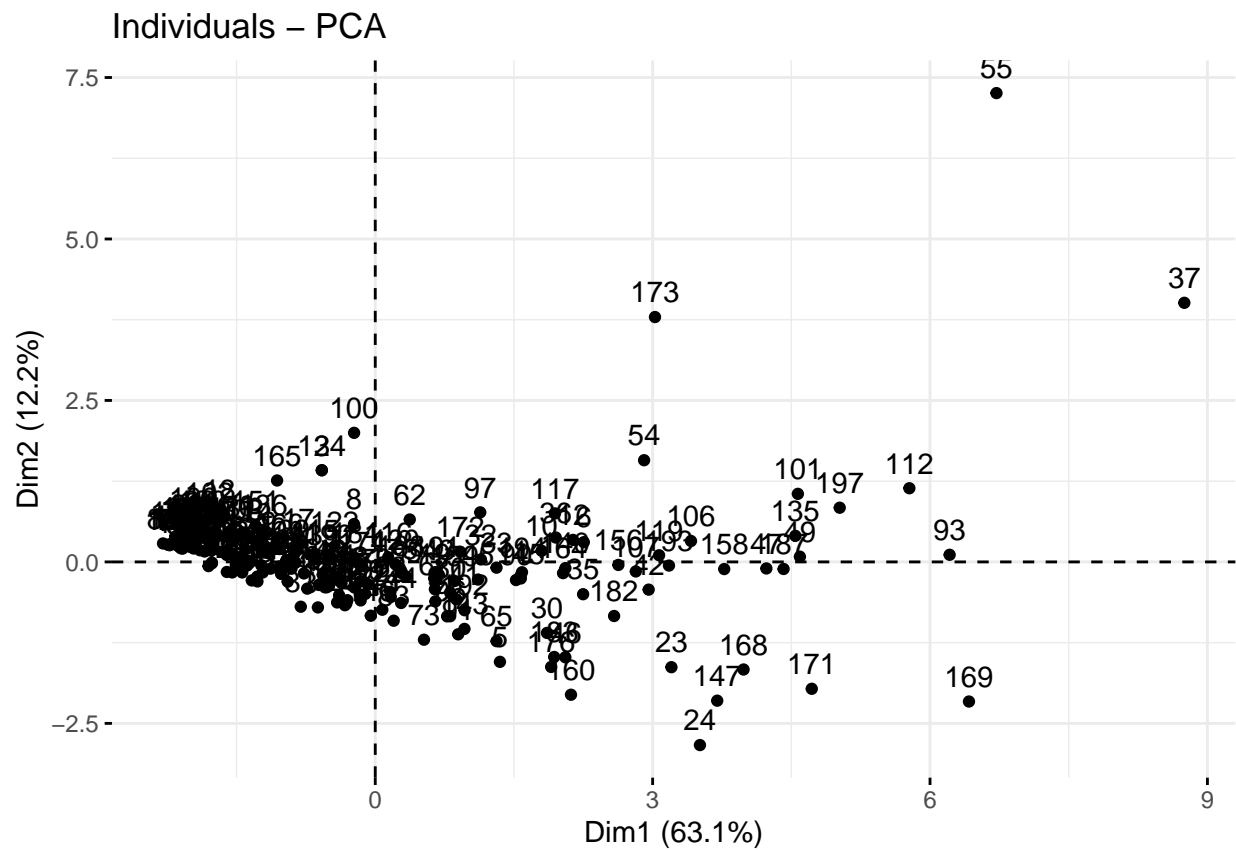
```
# Quality of individuals
head(ind$cos2)
```

```
##          Dim.1      Dim.2      Dim.3      Dim.4      Dim.5
## 1  0.63477606  0.0120601728  0.0026161756  0.003822273  0.20119323
## 2  0.05299496  0.3196599636  0.5430120851  0.051472409  0.01118200
## 3  0.87953016  0.0008739962  0.0385260264  0.002245031  0.01701061
## 4  0.94002261  0.0272645324  0.0005193995  0.009954769  0.01327390
## 5  0.34941785  0.4597750354  0.0757002063  0.061941650  0.00310777
## 6  0.68177783  0.0124892497  0.0544830511  0.043691120  0.15310469
```

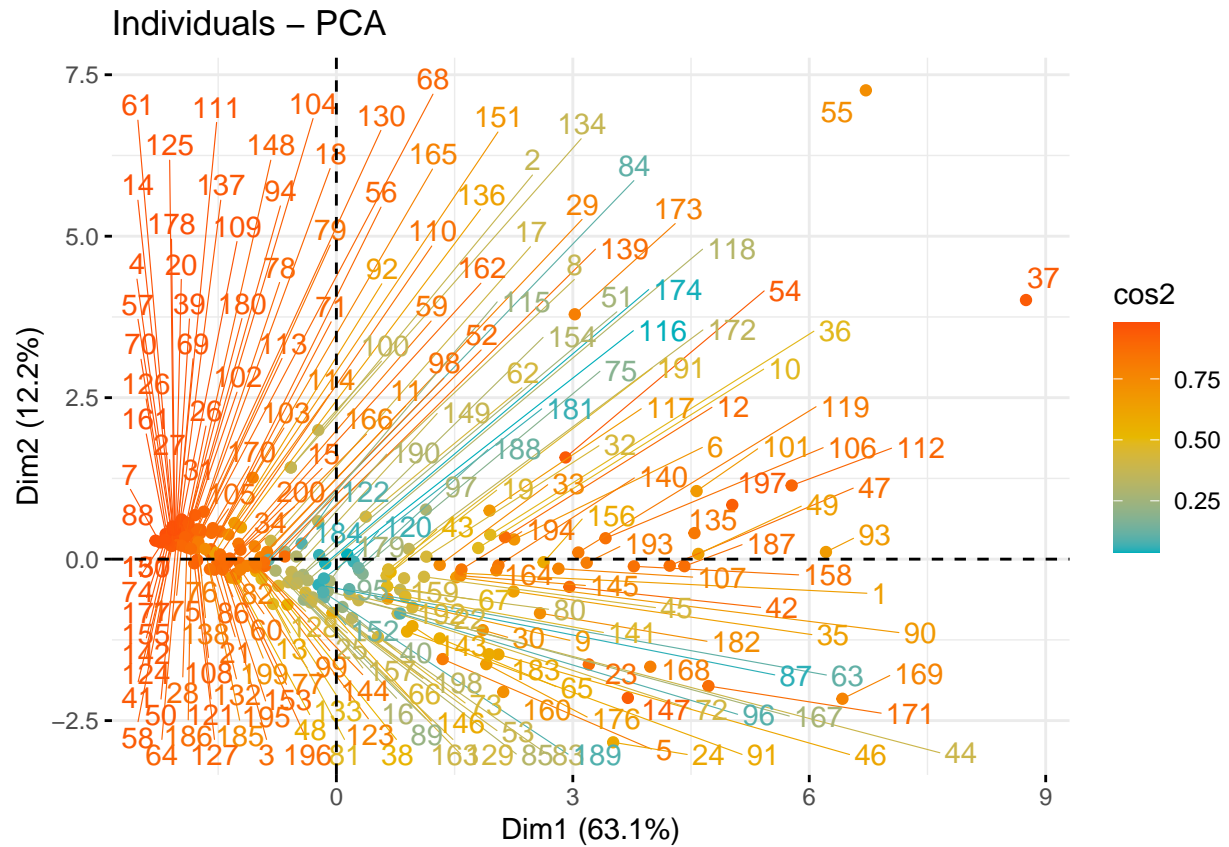
```
# Contributions of individuals
head(ind$contrib)
```

```
##          Dim.1      Dim.2      Dim.3      Dim.4      Dim.5
## 1  0.27979431  0.027567216  0.008393938  0.017579455  1.14127426
## 2  0.03781613  1.182910377  2.820542054  0.383250199  0.10268818
## 3  0.25729023  0.001325877  0.082036534  0.006852677  0.06403991
## 4  0.49015428  0.073724800  0.001971408  0.054161540  0.08907405
## 5  0.20600390  1.405713362  0.324868750  0.381046914  0.02357966
## 6  0.57611400  0.054729684  0.335125522  0.385233521  1.66499284
```

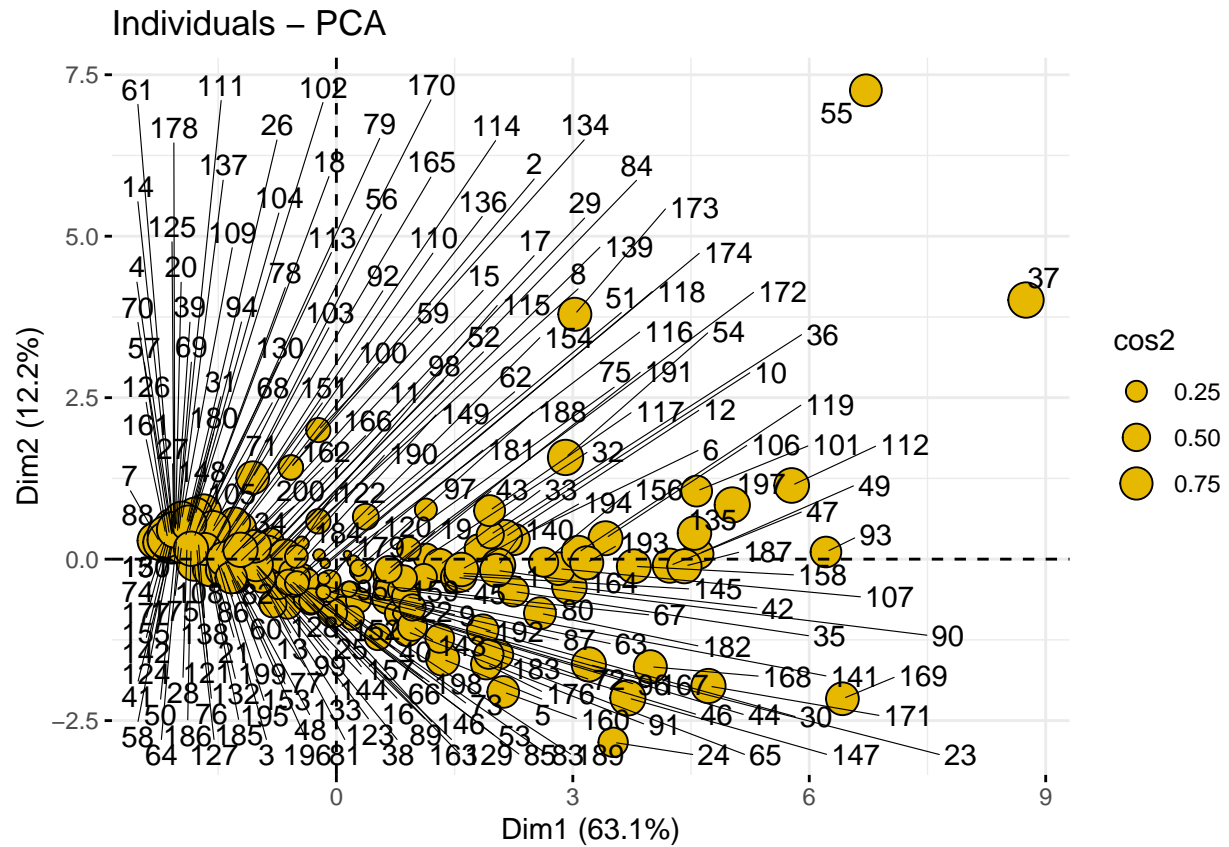
```
fviz_pca_ind(res.pca)
```



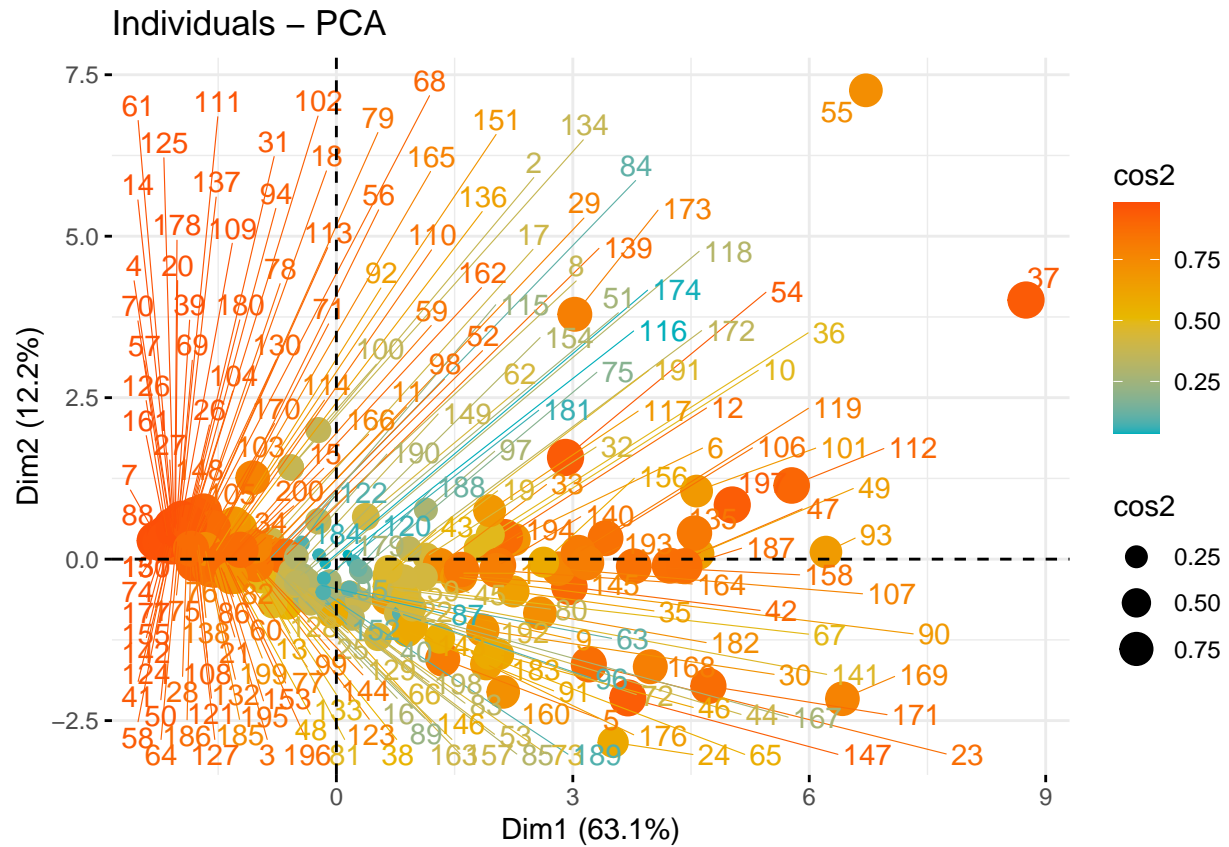
```
fviz_pca_ind(res.pca, col.ind = "cos2",
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE # Avoid text overlapping (slow if many points)
             )
```



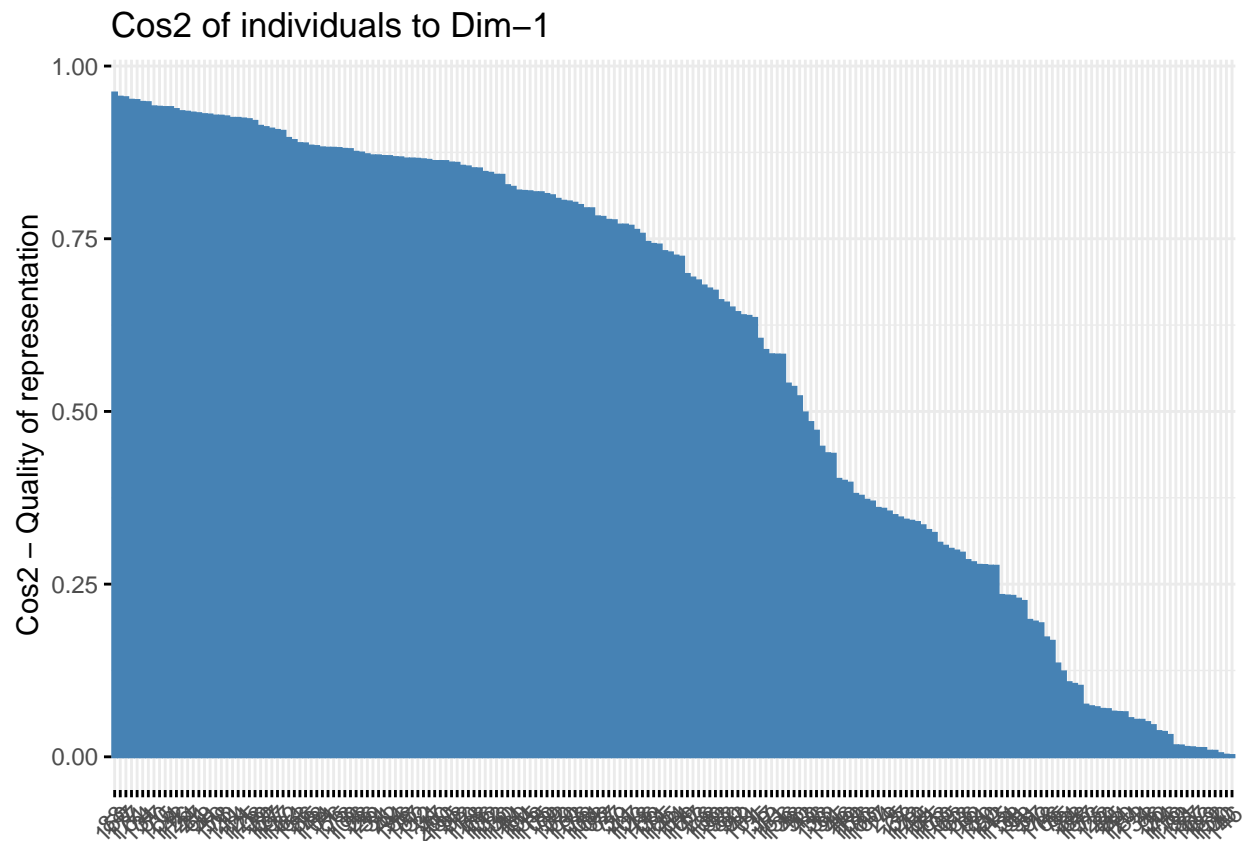
```
fviz_pca_ind(res.pca, pointsize = "cos2",
  pointshape = 21, fill = "#E7B800",
  repel = TRUE # Avoid text overlapping (slow if many points)
)
```



```
fviz_pca_ind(res.pca, col.ind = "cos2", pointsize = "cos2",
  gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
  repel = TRUE # Avoid text overlapping (slow if many points)
)
```

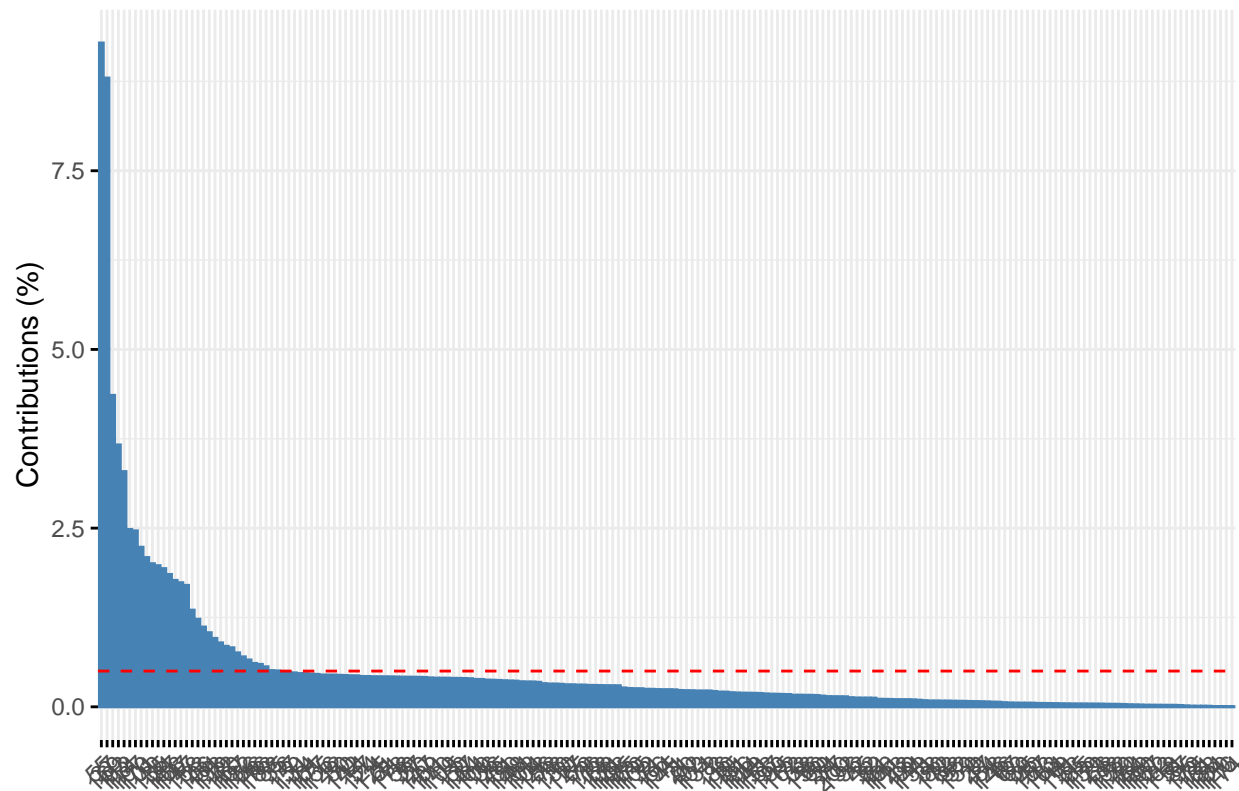


```
fviz_cos2(res.pca, choice = "ind")
```

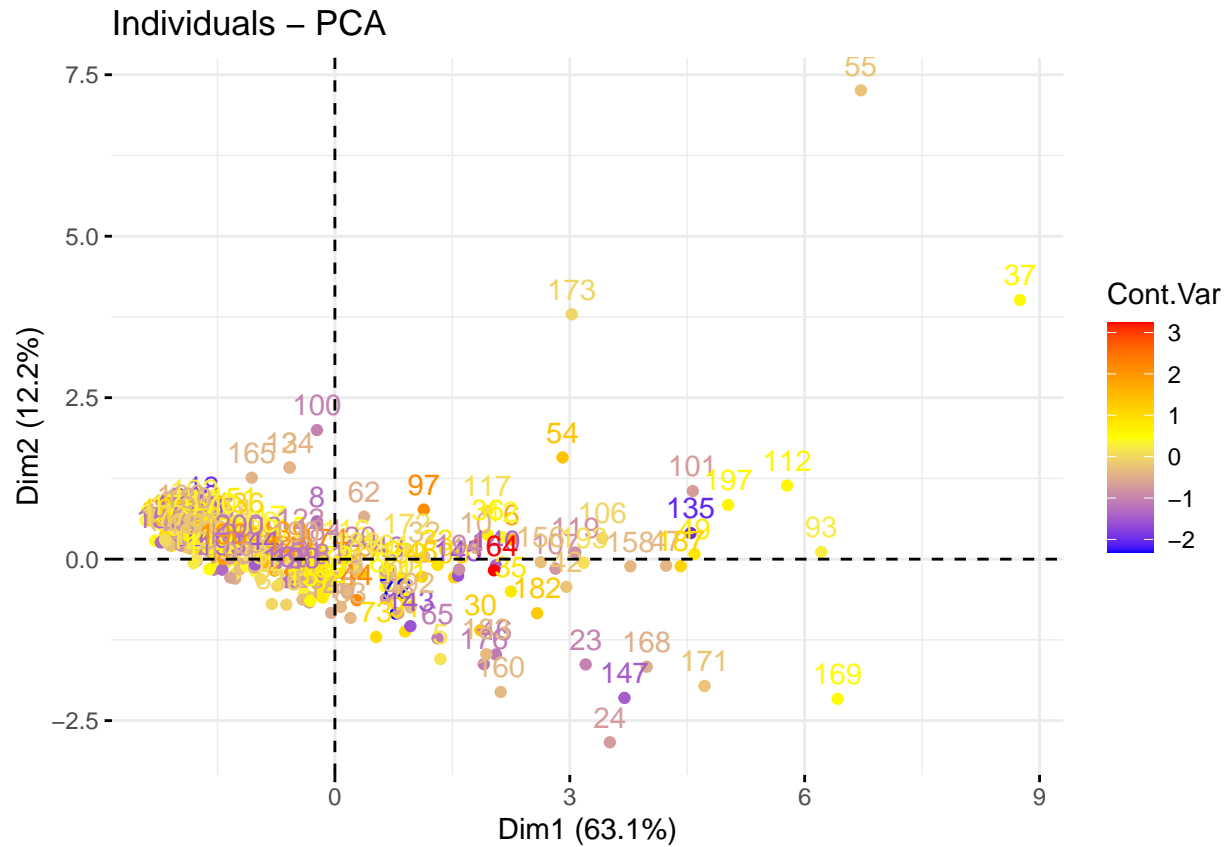


```
# Total contribution on PC1 and PC2  
fviz_contrib(res.pca, choice = "ind", axes = 1:2)
```

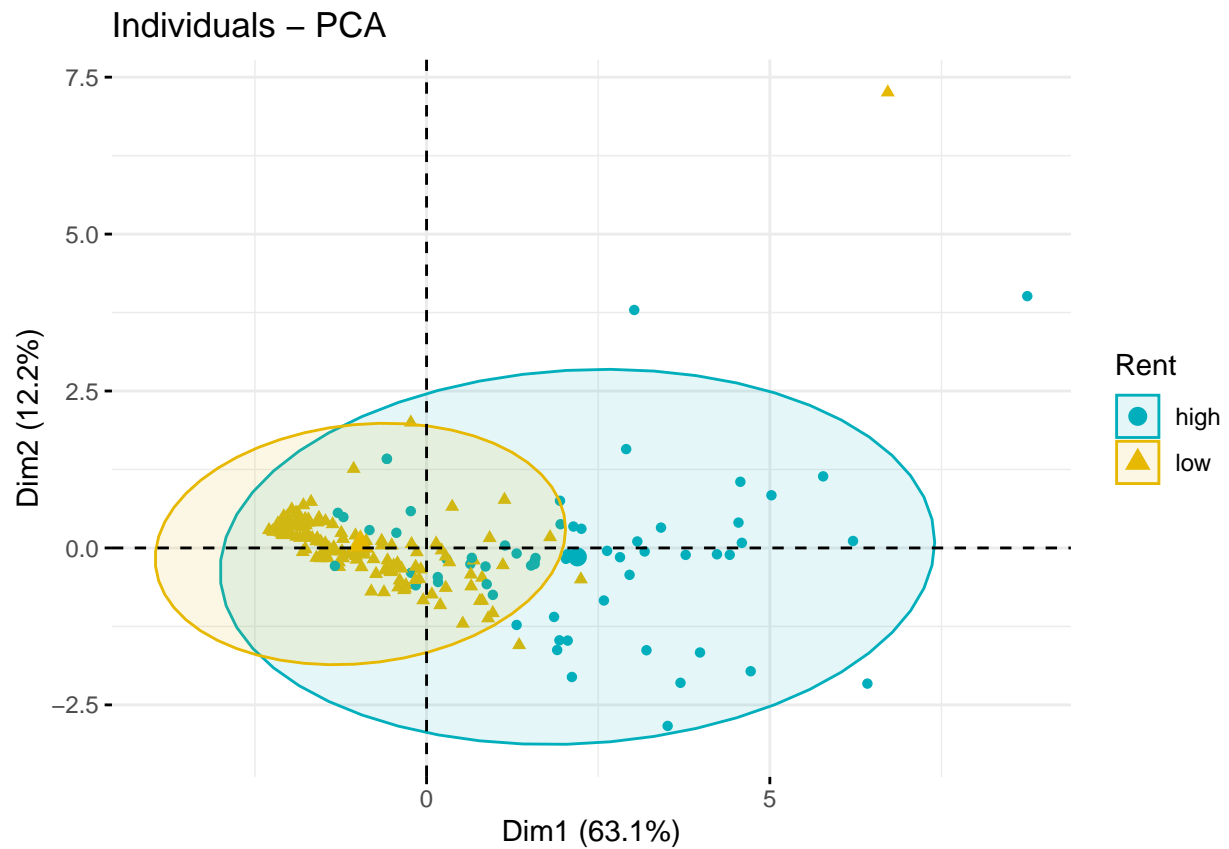

Contribution of individuals to Dim-1-2



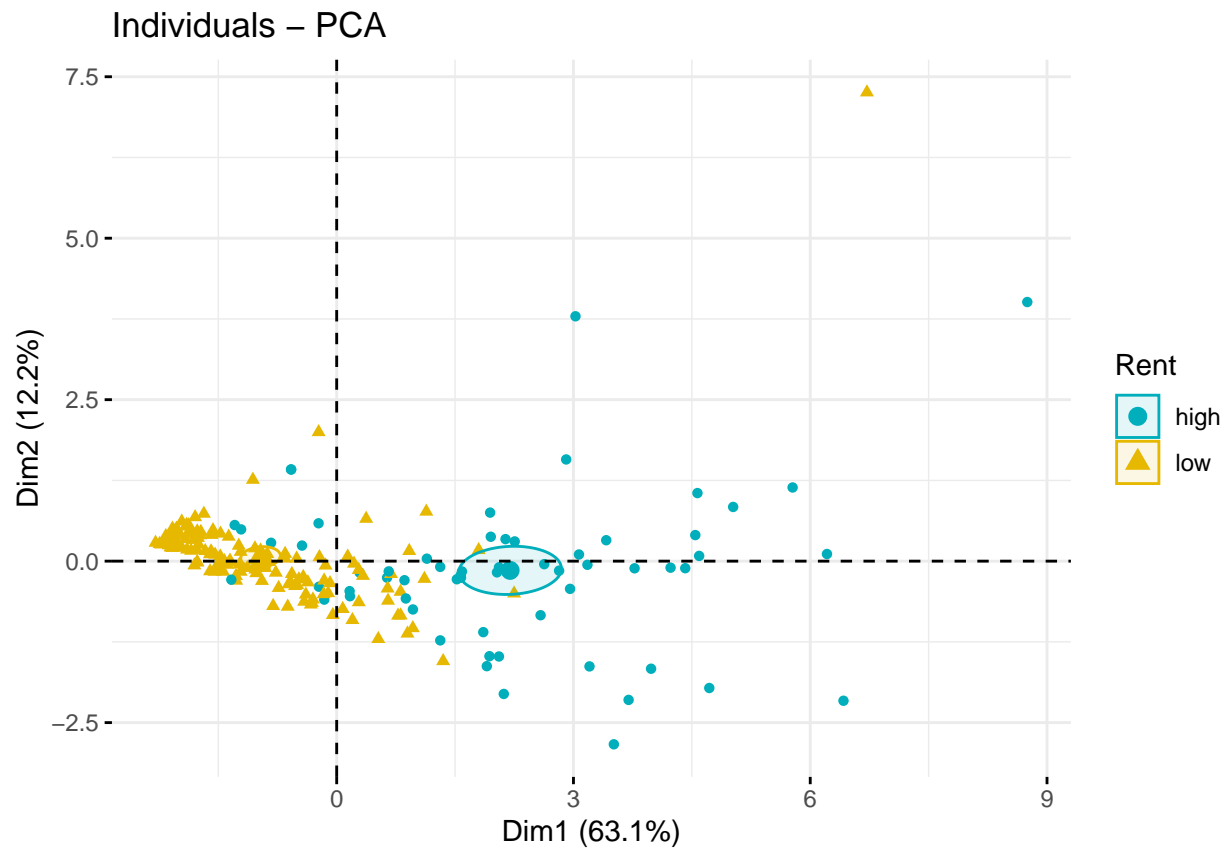
```
# Create a random continuous variable of length 23,  
# Same length as the number of active individuals in the PCA  
set.seed(123)  
my.cont.var <- rnorm(nrow(house_data))  
# Color individuals by the continuous variable  
fviz_pca_ind(res.pca, col.ind = my.cont.var,  
             gradient.cols = c("blue", "yellow", "red"),  
             legend.title = "Cont.Var")
```



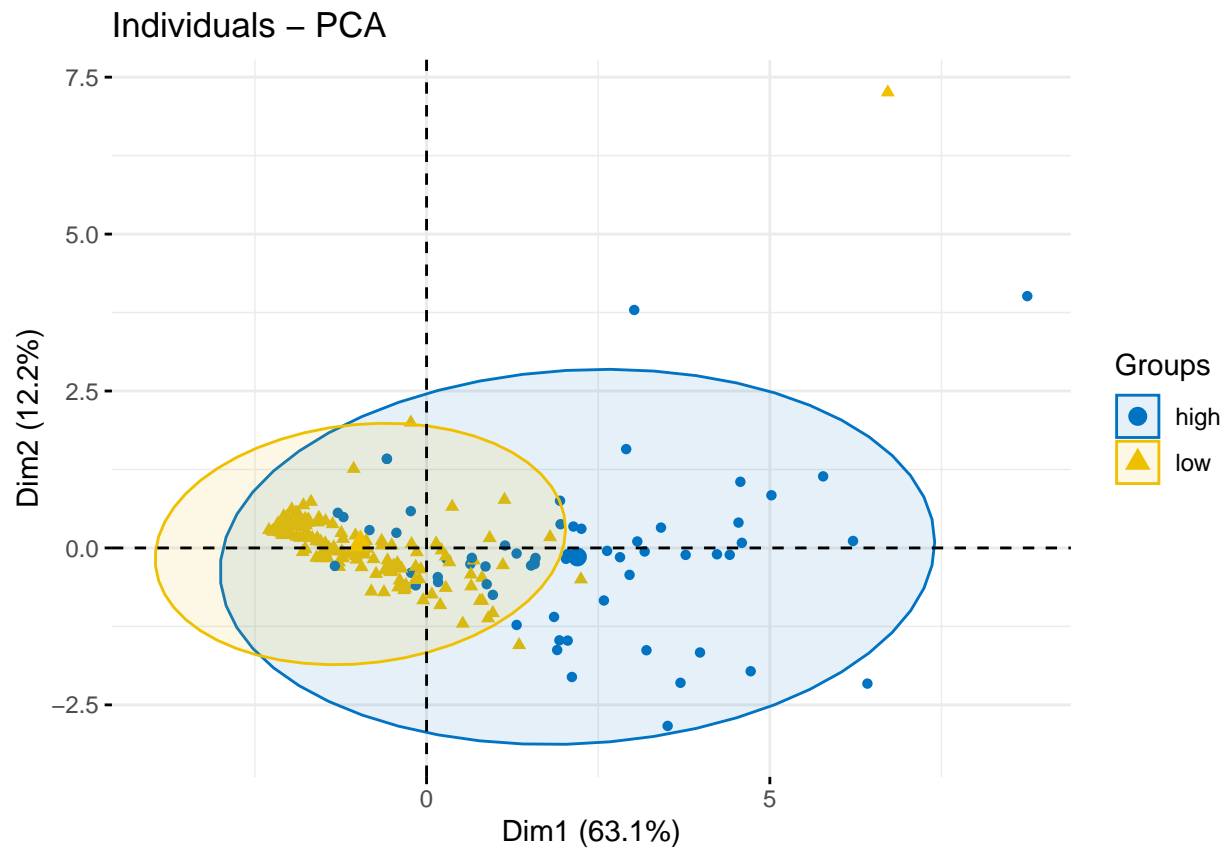
```
fviz_pca_ind(res.pca,
  geom.ind = "point", # show points only (nbut not "text")
  col.ind = house_data$rent, # color by groups
  palette = c("#00AFBB", "#E7B800", "#FC4E07"),
  addEllipses = TRUE, # Concentration ellipses
  legend.title = "Rent"
)
```



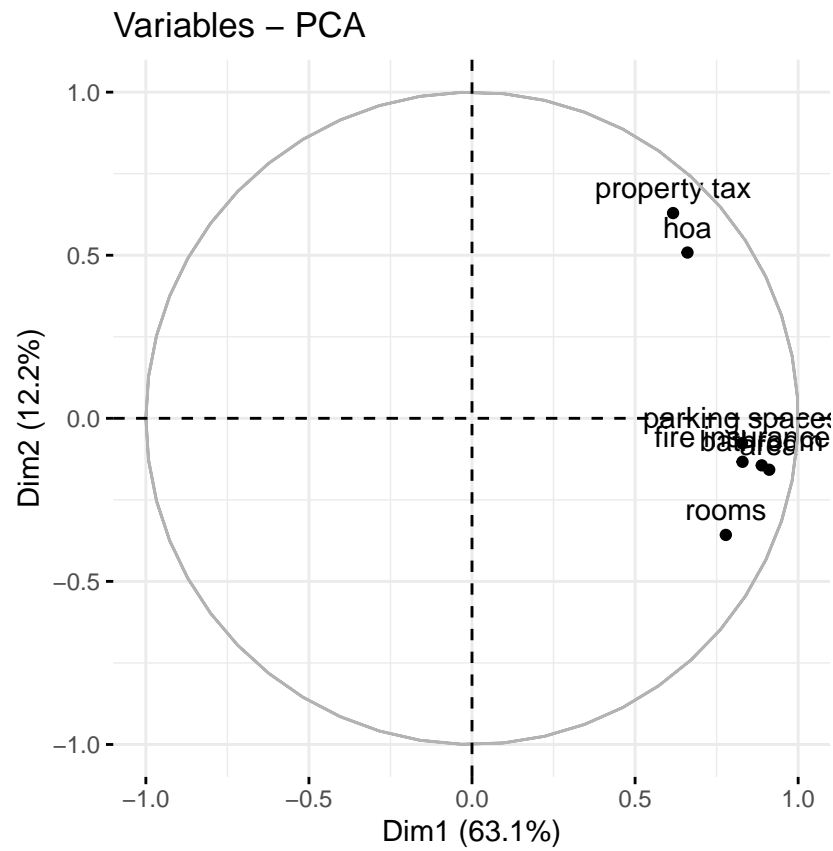
```
fviz_pca_ind(res.pca, geom.ind = "point", col.ind = house_data$rent,  
  palette = c("#00AFBB", "#E7B800", "#FC4E07"),  
  addEllipses = TRUE, ellipse.type = "confidence",  
  legend.title = "Rent"  
)
```



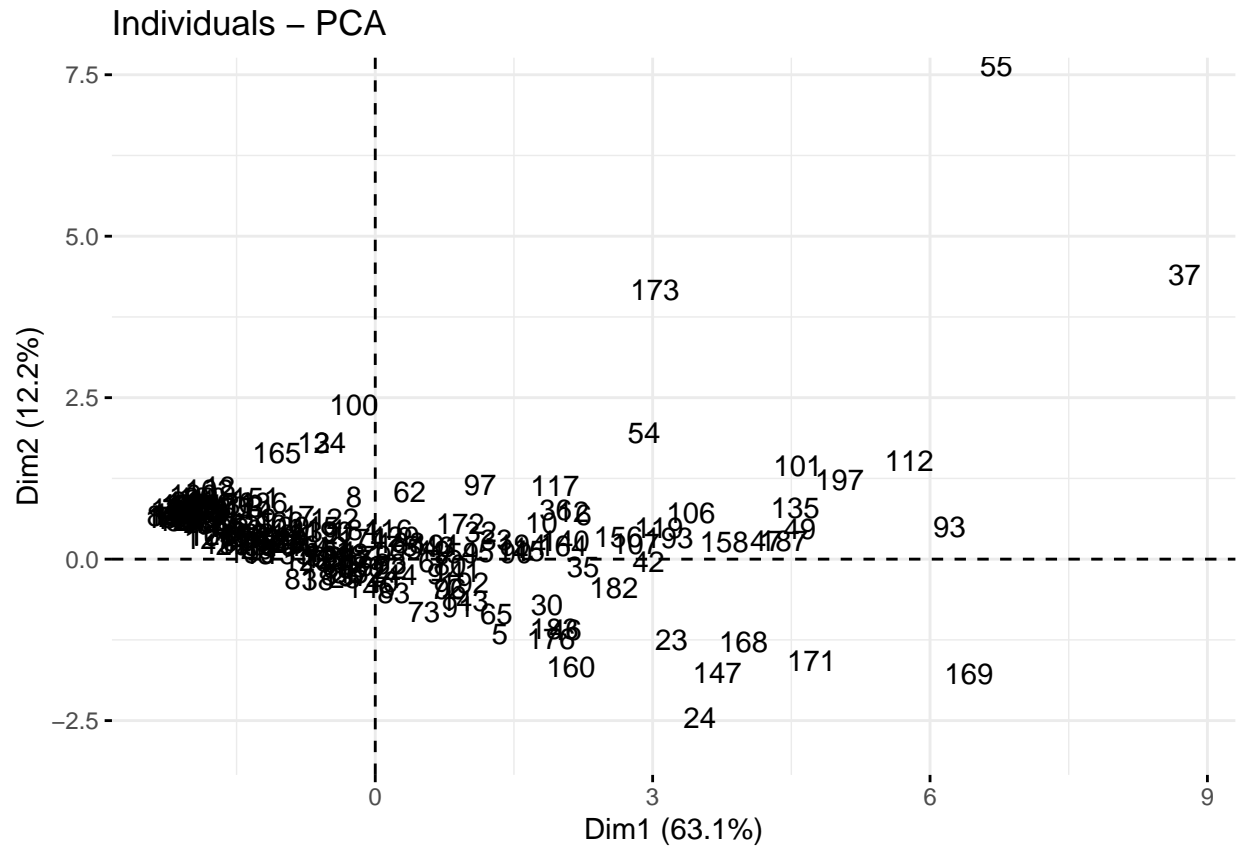
```
fviz_pca_ind(res.pca,  
  label = "none", # hide individual labels  
  habillage = house_data$rent, # color by groups  
  addEllipses = TRUE, # Concentration ellipses  
  palette = "jco"  
)
```



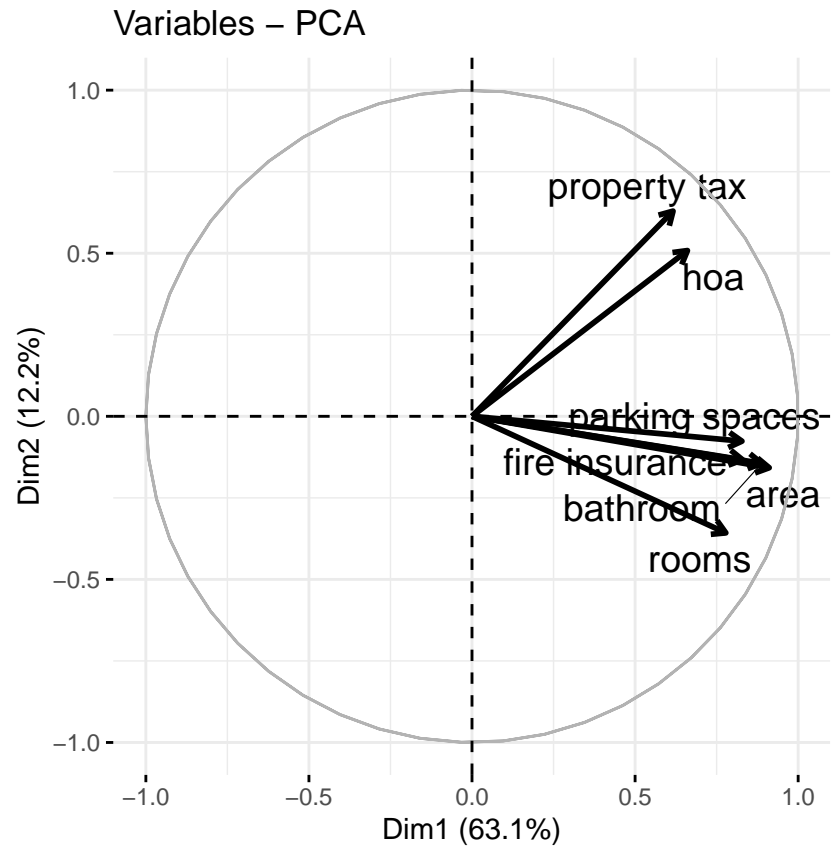
```
fviz_pca_var(res.pca, geom.var = c("point", "text"))
```



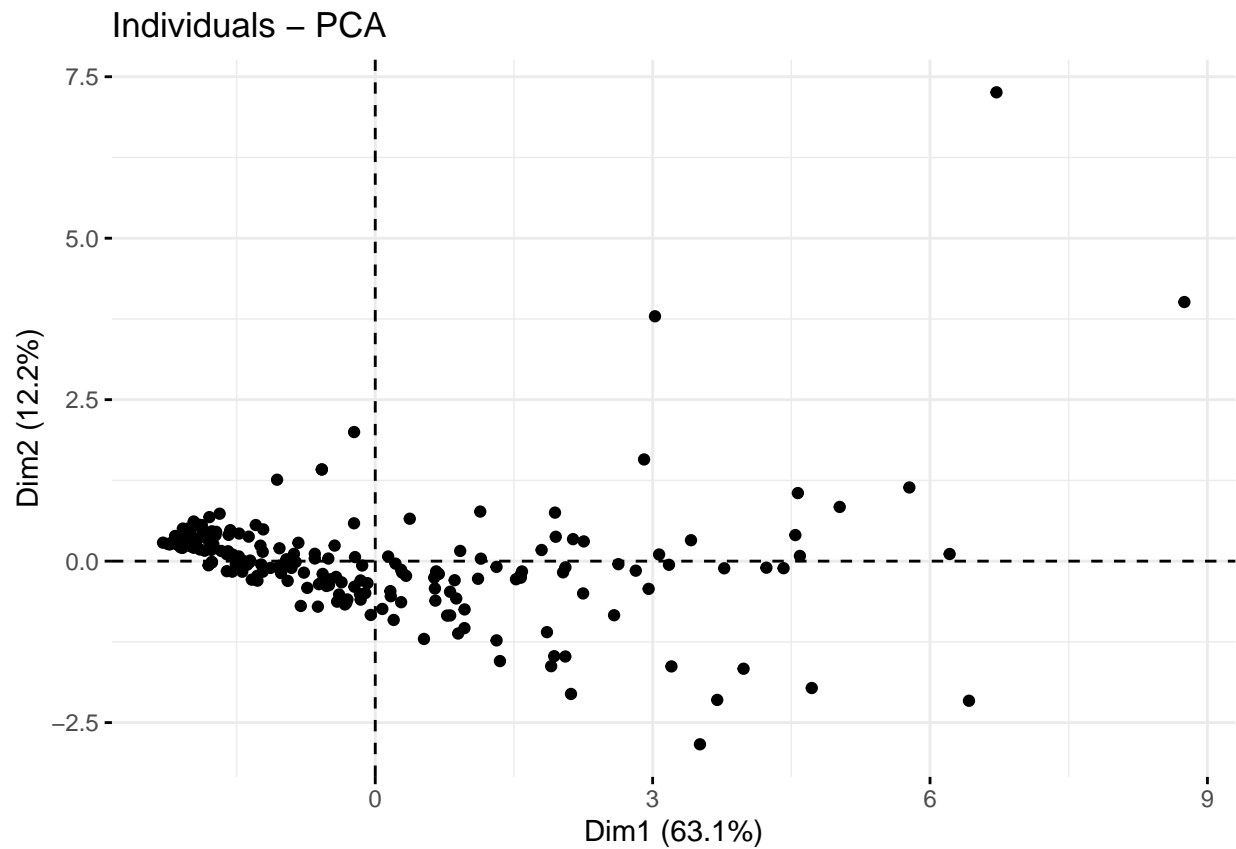
```
# Show individuals text labels only  
fviz_pca_ind(res.pca, geom.ind = "text")
```



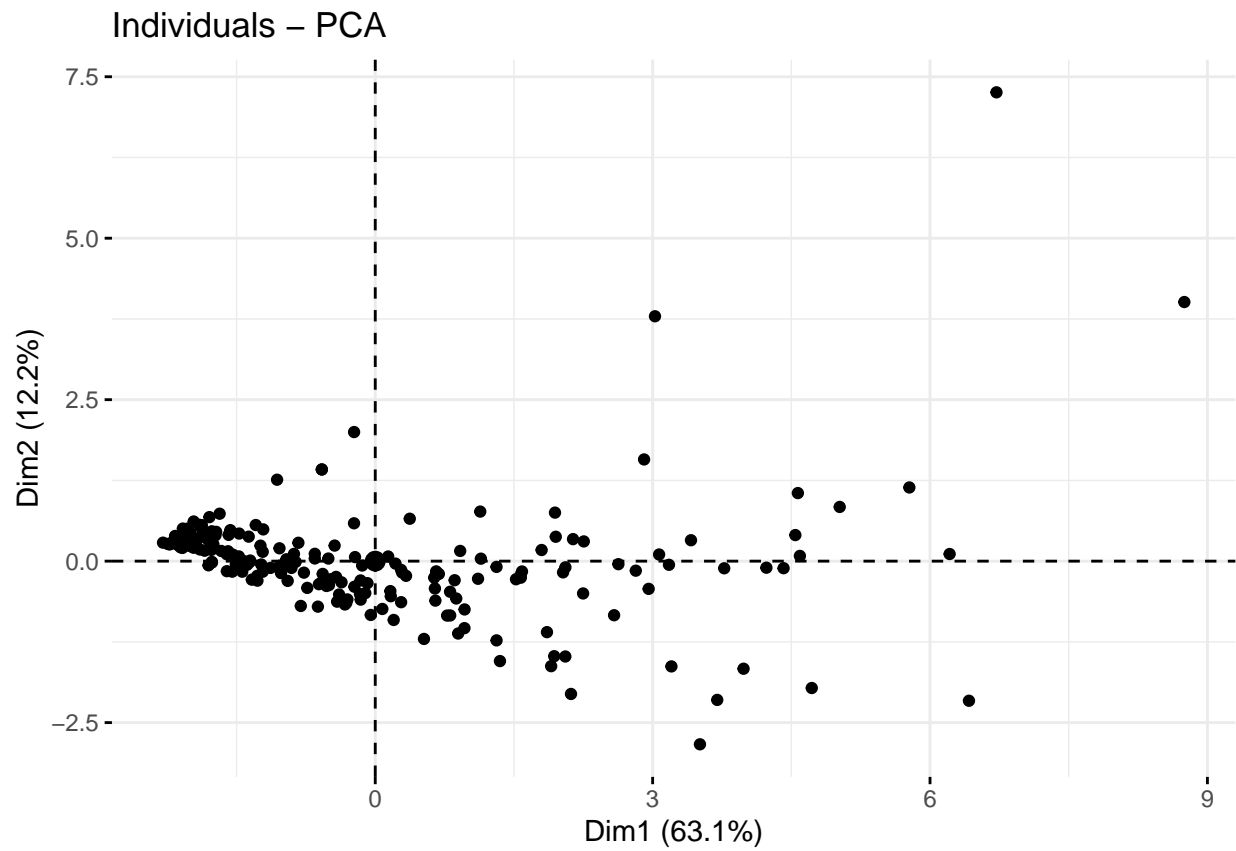
```
# Change the size of arrows and labels
fviz_pca_var(res.pca, arrowsize = 1, labelsize = 5,
              repel = TRUE)
```



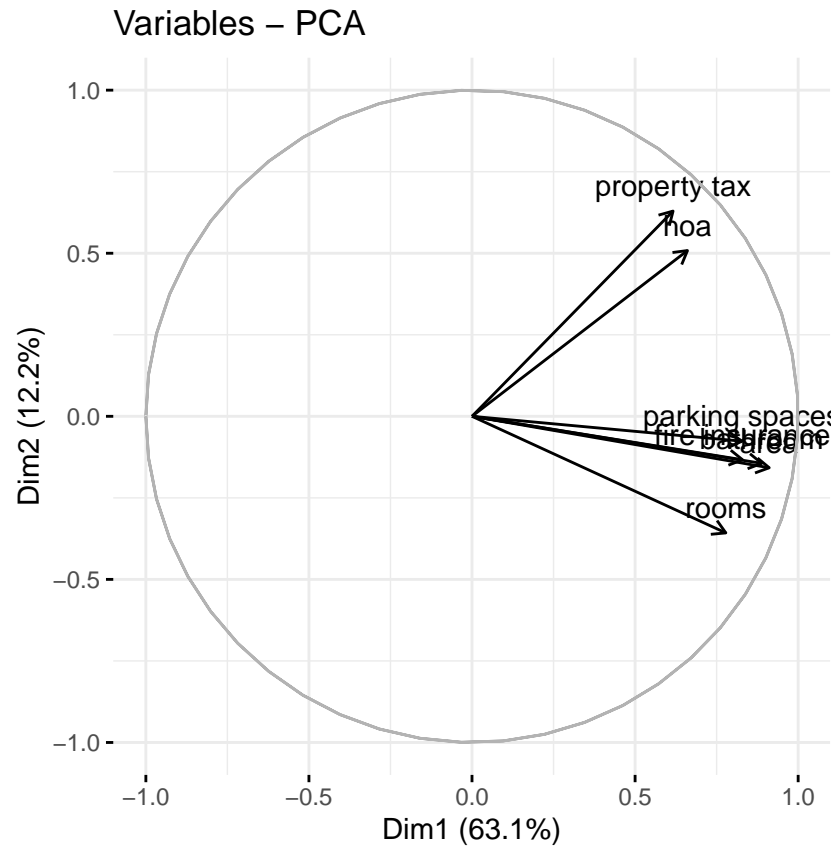
```
# Change points size, shape and fill color
# Change labelsize
fviz_pca_ind(res.pca,
  pointsize = 3, pointshape = 21, fill = "lightblue",
  labelsize = 5, repel = TRUE)
```

```
fviz_pca_ind(res.pca,  
  geom.ind = "point", # show points only (but not "text")  
  group.ind = house_data$rent, # color by groups  
  legend.title = "Rent",  
  mean.point = TRUE)
```



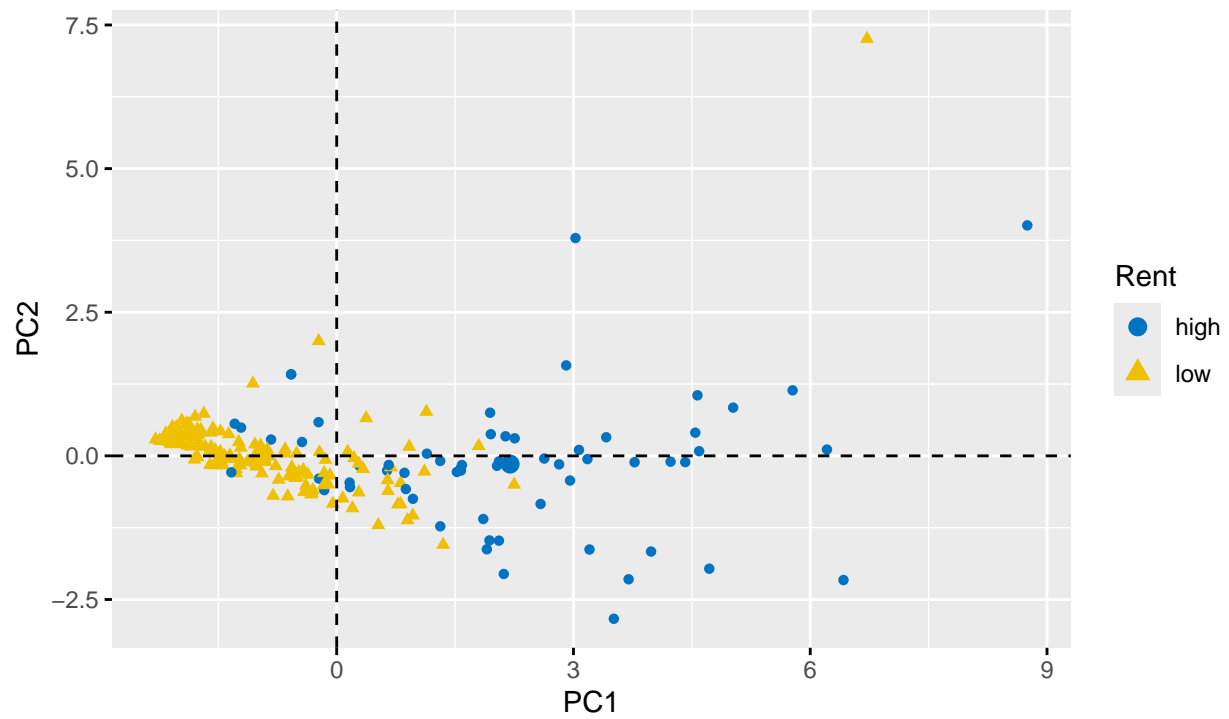
```
fviz_pca_var(res.pca, axes.linetype = "blank")
```



```
ind.p <- fviz_pca_ind(res.pca, geom = "point", col.ind = house_data$rent)
ggpubr::ggpar(ind.p,
  title = "Principal Component Analysis",
  subtitle = "Iris data set",
  caption = "Source: factoextra",
  xlab = "PC1", ylab = "PC2",
  legend.title = "Rent", legend.position = "top",
  ggtheme = theme_gray(), palette = "jco"
)
```

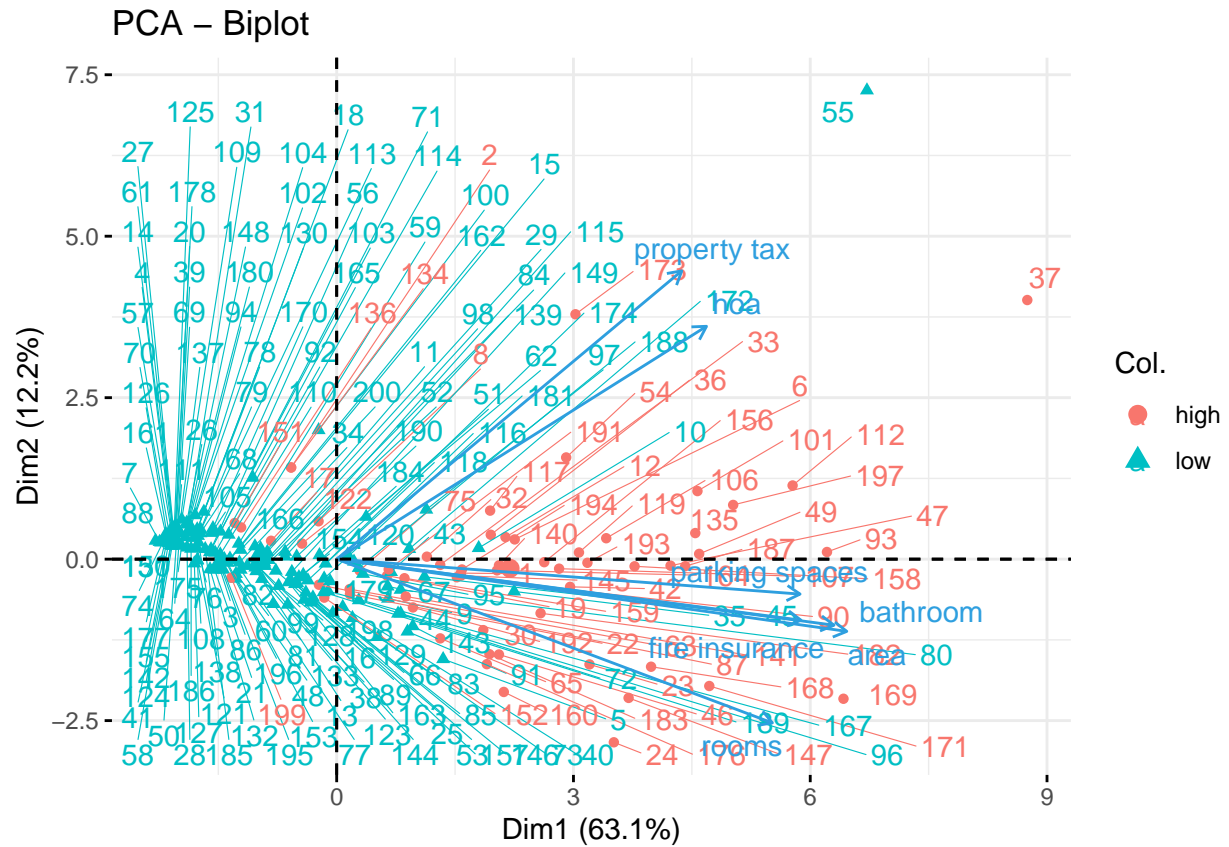
Principal Component Analysis

Iris data set

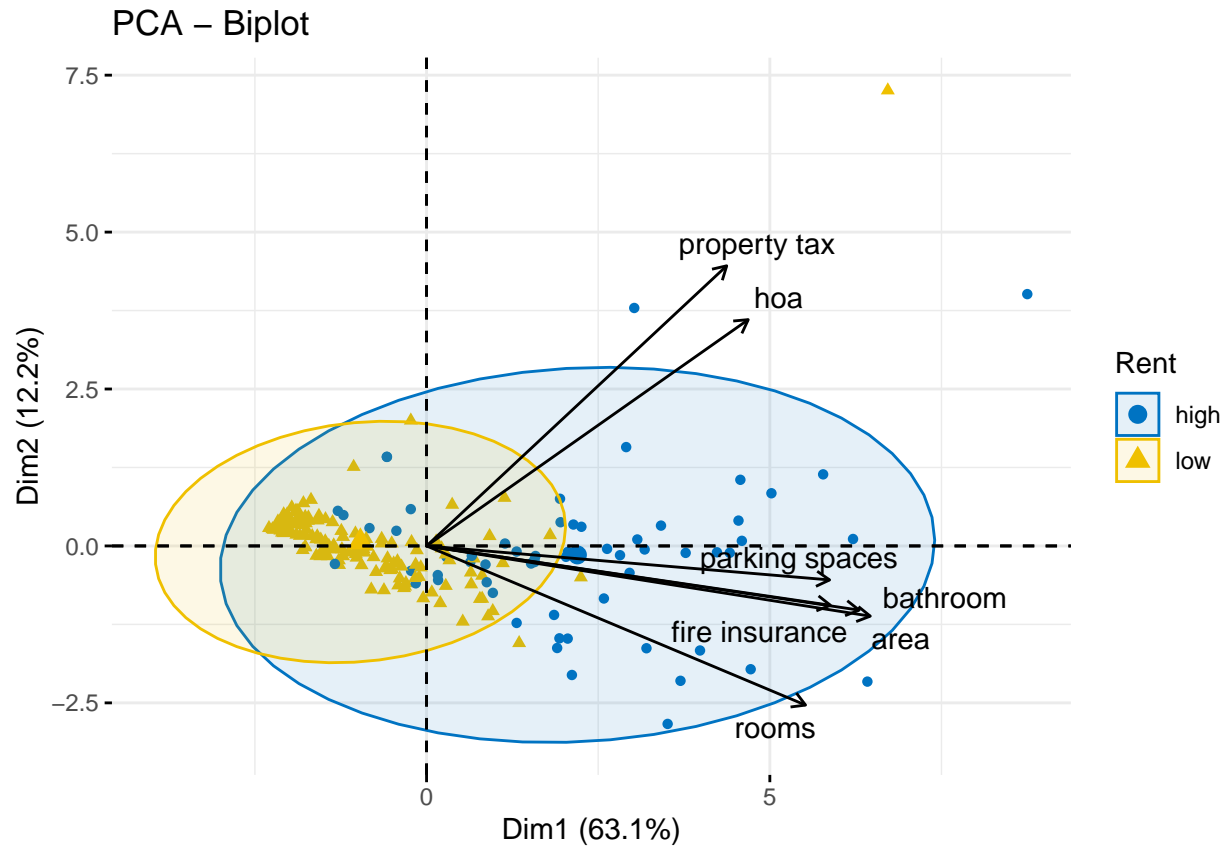


Source: factoextra

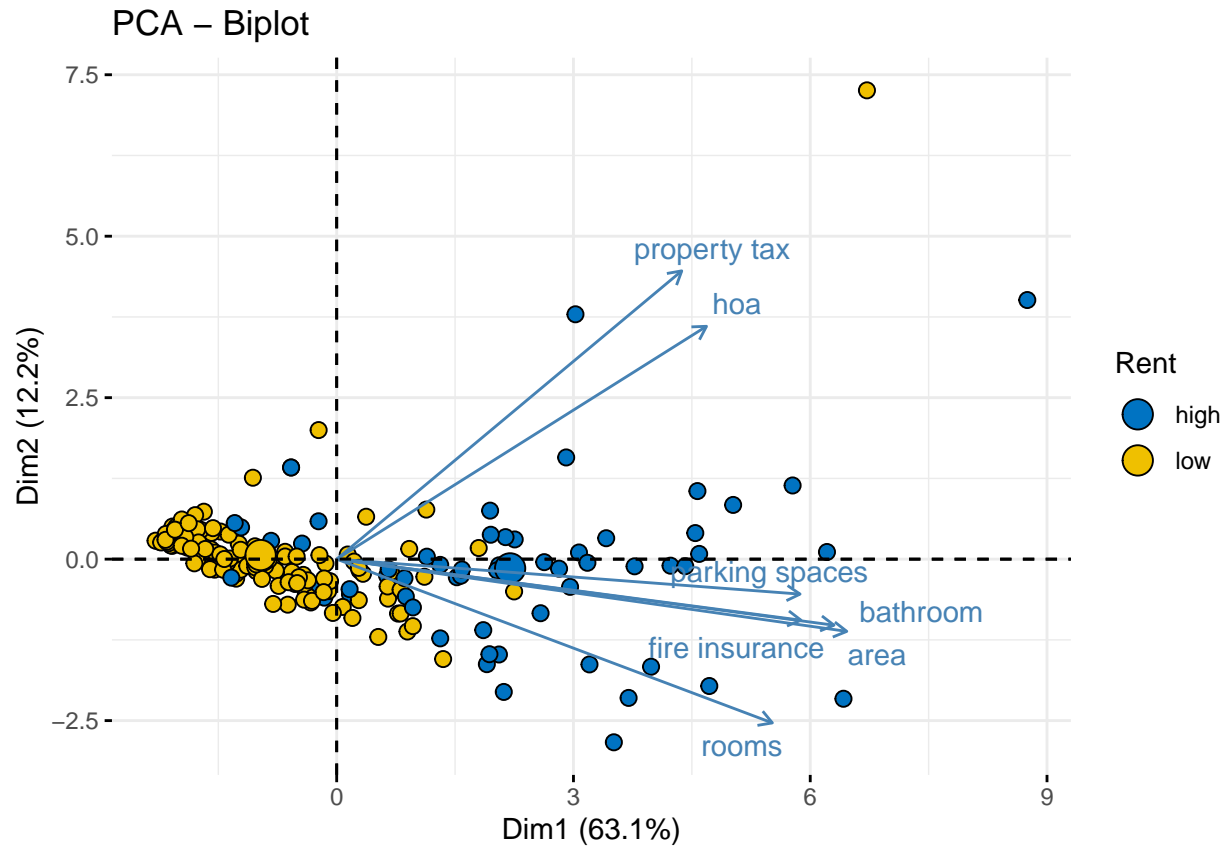
```
fviz_pca_biplot(res.pca, repel = TRUE, col.ind = house_data$rent,  
  col.var = "#2E9FDF", # Variables color  
)
```



```
fviz_pca_biplot(res.pca,
  col.ind = house_data$rent, palette = "jco",
  addEllipses = TRUE, label = "var",
  col.var = "black", repel = TRUE,
  legend.title = "Rent")
```



```
fviz_pca_biplot(res.pca,
  # Fill individuals by groups
  geom.ind = "point",
  pointshape = 21,
  pointsize = 2.5,
  fill.ind = house_data$rent,
  col.ind = "black",
  # Color variable by groups
  legend.title = list(fill = "Rent", color = "Clusters"),
  repel = TRUE      # Avoid label overplotting
)+
ggpubr::fill_palette("jco")+      # Individual fill color
ggpubr::color_palette("npg")      # Variable colors
```



```
fviz_pca_biplot(res.pca,
  # Individuals
  geom.ind = "point",
  fill.ind = house_data$rent, col.ind = "black",
  pointshape = 21, pointsize = 2,
  palette = "jco",
  addEllipses = TRUE,
  # Variables
  alpha.var = "contrib", col.var = "contrib",
  gradient.cols = "RdYlBu",

  legend.title = list(fill = "Rent", color = "Contrib",
    alpha = "Contrib")
)
```