348HWw5

Shay Lebovitz

10/12/2020

#5 *a)*

```r
paper <- read.table('/Users/shaylebovitz/R/paper.txt', header = TRUE)
(cor_mat <- cor(paper))
```

```
##           bl        em        sf        bs
## bl 1.0000000 0.9138256 0.9838790 0.9875554
## em 0.9138256 1.0000000 0.9422199 0.8746665
## sf 0.9838790 0.9422199 1.0000000 0.9745114
## bs 0.9875554 0.8746665 0.9745114 1.0000000
```

```r
apply(paper, 2, sd)
```

```
##        bl        em        sf        bs
## 2.8814703 0.7164910 1.4628895 0.6930166
```

`bl` has a high standard deviation, `sf` is medium, and `em` and `bs` have low standard deviations. All of the variables are very highly correlate with each other, the lowest correlation being `bs` and `em` with a correlation of 0.8747, which is still very high.

b) I think that the PC analysis should be based on the correlation matrix because there is a significant spread in the variable standard deviations, and they are also measured in different units.

c)

```r
paper_pca <- prcomp(paper, scale = TRUE)
paper_pca$rotation
```

```
##          PC1         PC2         PC3        PC4
## bl 0.5061685  0.26110200 -0.56517738 -0.5968196
## em 0.4854922 -0.81904792 -0.19350510  0.2366720
## sf 0.5080684  0.02020866  0.80019598 -0.3180323
## bs 0.4999573  0.51046828 -0.05307262  0.6976017
```

```r
summary(paper_pca)
```

```
## Importance of components:
##                           PC1     PC2     PC3    PC4
## Standard deviation     1.9595 0.37457 0.11227 0.0871
## Proportion of Variance 0.9599 0.03508 0.00315 0.0019
## Cumulative Proportion  0.9599 0.99495 0.99810 1.0000
```

Based on this analysis, just using the first principal componenet explaines 96% of the variance. By the second PC, it is already not 'pulling its weight' in that it explains less than 1/4 of the variance. For these reasons, I would just take the first PC in this case.

d)

```r
paper_sd <- apply(paper, 2, sd)
10*paper_pca$rotation[,1]/paper_sd
```

```
##       bl       em       sf       bs
## 1.756633 6.775970 3.473047 7.214218
```

Here we see most weight is on `em` and `bs`, which have the smallest standard deviations.

```r
paper_pca_scores <- predict(paper_pca)
head(paper_pca_scores)
```

```
##             PC1        PC2          PC3          PC4
## [1,] -0.3968987 0.15425573 -0.021786814 -0.009607294
## [2,] -0.5310917 0.16707751 -0.028802579 -0.049526457
## [3,] -0.9084521 0.25320477  0.035755303 -0.104024036
## [4,] -1.6010530 0.17423169  0.012458192 -0.025313914
## [5,] -1.1137187 0.08777323  0.014111553 -0.178803281
## [6,] -0.7434306 0.13672636 -0.004906303 -0.053276700
```

```r
cor(paper, paper_pca_scores)
```

```
##          PC1          PC2          PC3         PC4
## bl 0.9918199  0.097801441 -0.063450892 -0.05198256
## em 0.9513052 -0.306792239 -0.021724279  0.02061396
## sf 0.9955425  0.007569592  0.089835776 -0.02770039
## bs 0.9796492  0.191207016 -0.005958315  0.06076061
```

We see that the first principal component is extremely highly correlated with all the varaibles, and the rest of the PCs are essentailly non-correlated with any. I should also note that because the PC1 coefficients are so similar for every variable, it is essentailly an average of the variables.

#6 *a)*

```r
emplmnt <- read.table('/Users/shaylebovitz/R/employment.txt', header = TRUE)
emplmnt_pca <- prcomp(~AGR + MAN + CON + SER + FIN + SPS + TC,
                      data = emplmnt, scale = TRUE)
emplmnt_pca$rotation
```

```
##            PC1         PC2         PC3         PC4          PC5         PC6
## AGR  0.5982957 -0.06475831  0.03837215 -0.06426076  0.088358609  0.02948154
## MAN -0.1166513  0.41333890  0.29670088  0.82325922 -0.114998870 -0.02958918
## CON -0.2660980 -0.23118320  0.72664207 -0.29334483 -0.473901175 -0.07079849
## SER -0.4377825 -0.37921662  0.19288538  0.11736096  0.688108273  0.30824232
## FIN -0.1286410 -0.55516778 -0.43075986  0.32287232 -0.510847542  0.32195688
## SPS -0.5172193  0.10225300 -0.37282378 -0.11254728 -0.005356358 -0.61880152
## TC  -0.2861972  0.55591317 -0.14398285 -0.31839692 -0.141063088  0.64160581
##            PC7
## AGR -0.7896647
## MAN -0.1888280
## CON -0.1791411
## SER -0.2122653
## FIN -0.1442851
## SPS -0.4329204
## TC  -0.2353450
```

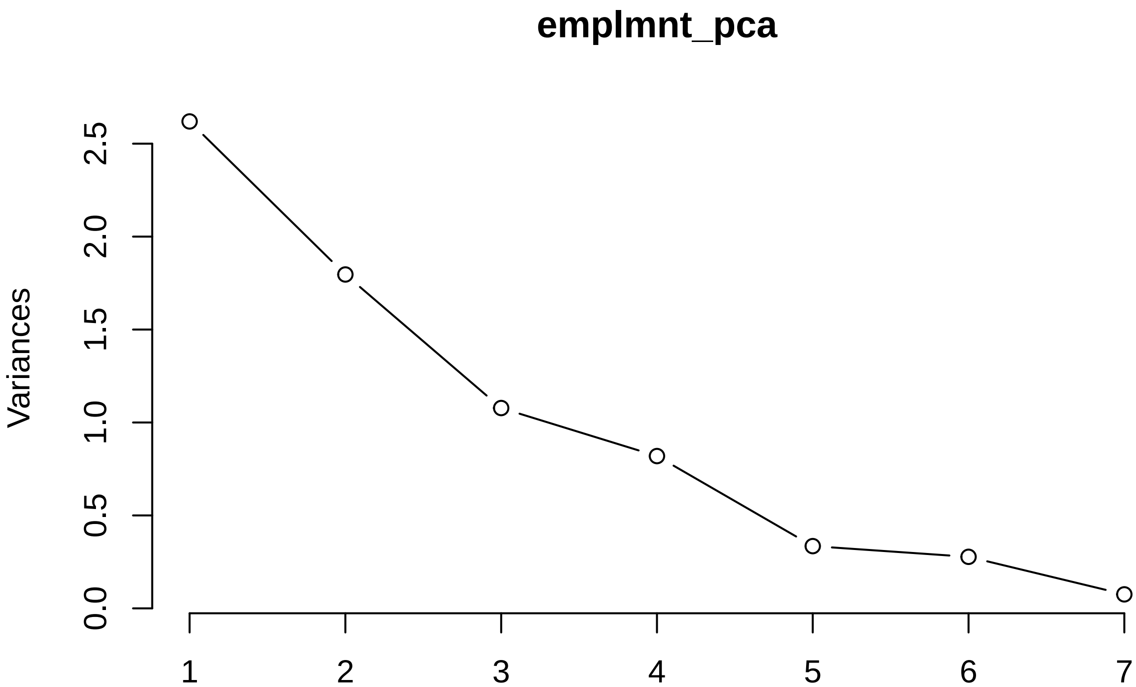
```r
emplmnt <- emplmnt[,-8]
```

b)

```r
summary(emplmnt_pca)
```

```
## Importance of components:
##                           PC1    PC2    PC3    PC4     PC5     PC6     PC7
## Standard deviation     1.6185 1.3402 1.0381 0.9051 0.57866 0.52653 0.27442
## Proportion of Variance 0.3742 0.2566 0.1540 0.1170 0.04784 0.03961 0.01076
## Cumulative Proportion  0.3742 0.6308 0.7848 0.9018 0.94964 0.98924 1.00000
```

```r
plot(emplmnt_pca, type = 'l')
```



I would choose the first 4 principal components, for a few reasons. Firstly, using the first four accounts for over 90% of the error, which is should be adequate for any analysis. Second, based on the scree plot, we see that the variance significantly plateaus after `PC4`, meaning they are not very useful. Choosing 3 PCs would work well too, as `PC4` doesn't 'pull its weight', as its variance is less than 1/7.

c)

```r
emplmnt_sd <- apply(emplmnt, 2, sd)
100*emplmnt_pca$rotation[,1]/emplmnt_sd
```

```
##       AGR       MAN       CON       SER       FIN       SPS        TC
##  4.861465 -1.233519 -9.736175 -8.483897 -3.226770 -5.923220 -23.204533
```

Here we see that only `AGR` is positive, the rest are negative.

```r
emplmnt_pca_scores <- predict(emplmnt_pca)
head(emplmnt_pca_scores)
```

```
##          PC1         PC2        PC3         PC4          PC5         PC6
## 1 -1.1931943  0.07086769 -0.9785269  0.20434044 -0.001740173 -0.27339172
## 2 -0.8723157  0.23293159 -1.0857976  0.07703399 -0.386479020 -0.23204432
## 3 -0.8556792 -0.45515661 -0.7374718  0.32236092 -0.286008258 -0.11585267
## 4 -0.7877522 -0.80746205  0.3900067  0.71575214 -0.516455050 -0.29606174
## 5  0.7197939  0.07827075  0.2997098 -0.14216014  0.679818185  0.83251243
## 6  0.1263532 -0.70803017 -0.0927552  0.37419949  0.231938444  0.05303738
##            PC7
```

```
## 1  0.002028068
## 2 -0.113200353
## 3  0.019824050
## 4  0.285948904
## 5 -0.358136025
## 6 -0.019239188
```

```r
cor(emplmnt, emplmnt_pca_scores)
```

```
##            PC1         PC2         PC3         PC4         PC5         PC6
## AGR  0.9683415 -0.08678933  0.03983426 -0.05816447  0.05112959  0.01552303
## MAN -0.1888001  0.55395834  0.30800618  0.74515825 -0.06654525 -0.01557971
## CON -0.4306797 -0.30983260  0.75432957 -0.26551578 -0.27422765 -0.03727781
## SER -0.7085509 -0.50822754  0.20023496  0.10622716  0.39818073  0.16230007
## FIN -0.2082055 -0.74403794 -0.44717326  0.29224206 -0.29560704  0.16952124
## SPS -0.8371194  0.13703985 -0.38702962 -0.10187015 -0.00309951 -0.32582004
## TC  -0.4632101  0.74503691 -0.14946908 -0.28819123 -0.08162757  0.33782728
##             PC7
## AGR -0.21669623
## MAN -0.05181732
## CON -0.04915909
## SER -0.05824890
## FIN -0.03959406
## SPS -0.11880006
## TC  -0.06458231
```

We see that `AGR` is highly correlated with `PC1`, and the rest are negatively correlated. `PC2` shows strong correlation with `TC` and strong negative correlation with `FIN`. `PC3` really only shows strong correlation with `CON`.

#7 *a)*

```r
pollution <- read.table('/Users/shaylebovitz/R/pollution.txt', header = TRUE)
(pollution_sd <- apply(pollution, 2, sd))
```

```
##        CO        NO       NO2        O3        HC
## 1.2337209 1.0873574 3.3709837 5.5658345 0.6917466
```

```r
cor(pollution)
```

```
##            CO         NO       NO2         O3        HC
## CO  1.0000000  0.5021525 0.5565838  0.4109288 0.1660323
## NO  0.5021525  1.0000000 0.2968981 -0.1339521 0.2347043
## NO2 0.5565838  0.2968981 1.0000000  0.1666422 0.4477678
## O3  0.4109288 -0.1339521 0.1666422  1.0000000 0.1544506
## HC  0.1660323  0.2347043 0.4477678  0.1544506 1.0000000
```

b)

```r
(pollution_pca_cov <- prcomp(pollution, scale = F))
```

```
## Standard deviations (1, .., p=5):
## [1] 5.6410664 3.3862185 1.1984182 0.7293890 0.5183157
##
## Rotation (n x k) = (5 x 5):
##             PC1        PC2        PC3          PC4         PC5
## CO  -0.10343698 0.18207274 0.62534826 -0.578141442  0.48046054
## NO   0.01778827 0.12836692 0.74801509  0.500285034 -0.41640584
```

```
## NO2 -0.16191213  0.95490887 -0.22085443 -0.004444182 -0.11461711
## O3  -0.98090512 -0.17661949 -0.01542742  0.054814906 -0.05820662
## HC  -0.02437144  0.08559241 -0.01995722  0.642217201  0.76107736
```

```
(pollution_pca_cor <- prcomp(pollution, scale = T))
```

```
## Standard deviations (1, .., p=5):
## [1] 1.4875002 1.0644756 0.9522246 0.7415103 0.4445961
##
## Rotation (n x k) = (5 x 5):
##            PC1         PC2        PC3         PC4        PC5
## CO  -0.5621560  0.10573277 -0.4645354 -0.07701329 -0.6716227
## NO  -0.4106208 -0.57912175 -0.3241413  0.45949883  0.4240304
## NO2 -0.5394732 -0.03035633  0.2155503 -0.71908494  0.3801343
## O3  -0.2689306  0.80581727 -0.1087780  0.34033332  0.3881693
## HC  -0.3898924 -0.05635245  0.7879370  0.38732397 -0.2719260
```

`O3` has the highest standard deviation, and in the unscaled `PC1`, carries practically all the weight of the principal componenet. Using the scaled version, we see that the coefficients of `PC1` are much more even across variables, with `O3` now being the smallest. Based on the correlation matrix, `O3` is the least correlated variable.

c)

```
pollution_pca_scaled <- prcomp(~I(CO/35) + I(NO/25) + I(NO2/5) + I(O3/7.5) +
                                I(HC/25), data = pollution, scale = FALSE)
pollution_pca_scaled$rotation
```

```
##                    PC1         PC2          PC3          PC4         PC5
## I(CO/35)  -0.026627863  0.01495777 -0.436121552  0.478471590  0.76153145
## I(NO/25)  -0.001081216  0.02398402 -0.897456221 -0.176704632 -0.40344938
## I(NO2/5)  -0.501023048  0.86458853  0.031632813  0.006518366 -0.02048055
## I(O3/7.5) -0.864945991 -0.50149247 -0.002999093 -0.006768663 -0.01785851
## I(HC/25)  -0.011581302  0.01389266 -0.057971489 -0.860088430  0.50651759
```

```
summary(pollution_pca_scaled)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5
## Standard deviation     0.7742 0.6378 0.04430 0.02637 0.01829
## Proportion of Variance 0.5939 0.4031 0.00194 0.00069 0.00033
## Cumulative Proportion  0.5939 0.9970 0.99898 0.99967 1.00000
```

I would say that the first 2 PCs are needed, as the first one only covers roughly 60% which is probably not adequate. The first two cover >99% of the variance, which is certainly adequate. `NO2` and `O3` are the only variables with significant `PC1` and `PC2` coefficients.

#8 *a)*

```
qbs <- read.table('/Users/shaylebovitz/R/QBs.txt', header = TRUE)
apply(qbs, 2, sd)
```

```
##       Comp         TD        Int        YPA       Rate
##  3.9040116  1.4322015  1.0264446  0.7053741 12.9562610
```

Because the standard deviations vary so much, I will perform the analysis based on the correlation matrix

```
qbs_pca <- prcomp(~Comp + TD + Int + YPA, data = qbs, scale = TRUE)
qbs_pca$rotation
```

```
##             PC1         PC2        PC3         PC4
```

```
## Comp -0.4782130  0.43465731  0.7630190  0.01368644
## TD   -0.5533956 -0.05905053 -0.3268963  0.76380962
## Int   0.4441221  0.82079809 -0.1946390  0.30192998
## YPA  -0.5175144  0.36589145 -0.5225480 -0.57030328
```

```r
summary(qbs_pca)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4
## Standard deviation     1.6890 0.7796 0.63236 0.37390
## Proportion of Variance 0.7131 0.1519 0.09997 0.03495
## Cumulative Proportion  0.7131 0.8651 0.96505 1.00000
```

b) From the principal components analysis, we see that about 71% of the total variation can be explained by `PC1`. That, along with the fact that `PC2` only explains about 15% of the variation and thus doesn't carry its weight, leads me to believe that just one PC will be adequate for the analysis.

c)

```r
qb_pca_scores <- predict(qbs_pca)
qb_pca_scores[,1]
```

```
##           1           2           3           4           5           6
## -3.65831935 -0.38348573 -2.38308495  0.35446489  0.11556998  1.79272556
##           7           8           9          10          11          12
##  0.71186252 -0.27533342 -0.68063062  0.12046717  0.21251653  2.09483279
##          13          14          15          16          17          18
## -2.58825500 -1.17325054  0.03052064  1.44379074 -0.21377860  0.63054126
##          19          20          21          22          23          24
##  2.44837145  0.63175117 -0.55726141 -1.91096316  1.69559675 -0.41841303
##          25          26          27          28          29          30
##  0.55075158  1.43170346 -4.18523072  1.32115054 -2.55659500  2.20824872
##          31          32          33          34          35          36
##  2.27635849 -0.17420932  0.62522057  1.22807276  1.08248146  1.02623722
##          37
## -2.87442540
```

```r
cor(qbs$Rate, qb_pca_scores[,1])
```

```
## [1] -0.9963963
```

We see that there is a very strong negative correlation between the first principal component and the quarterback rate. This is surprising as `Rate` has the highest standard deviation of all the variables, so I'm not exactly sure on how to analyze this. **#9)** *a)*

```r
properties <- read.table('/Users/shaylebovitz/R/properties.txt', header = T)
cor(properties)
```

```
##            BATH        LOT       SIZE         GAR      ROOM       BED
## BATH  1.0000000  0.4129558  0.7285916  0.22402204 0.5103104 0.4264014
## LOT   0.4129558  1.0000000  0.5715520  0.20466375 0.3921244 0.1516093
## SIZE  0.7285916  0.5715520  1.0000000  0.35888351 0.6788606 0.5743353
## GAR   0.2240220  0.2046638  0.3588835  1.00000000 0.5893871 0.5412988
## ROOM  0.5103104  0.3921244  0.6788606  0.58938707 1.0000000 0.8703883
## BED   0.4264014  0.1516093  0.5743353  0.54129880 0.8703883 1.0000000
## AGE  -0.1007485 -0.3527514 -0.1390869 -0.02016883 0.1242663 0.3135114
## Y     0.7097771  0.6476364  0.7077656  0.46146792 0.5284436 0.2815200
##              AGE          Y
```

```
## BATH -0.10074847  0.7097771
## LOT  -0.35275139  0.6476364
## SIZE -0.13908686  0.7077656
## GAR  -0.02016883  0.4614679
## ROOM  0.12426629  0.5284436
## BED   0.31351144  0.2815200
## AGE   1.00000000 -0.3974034
## Y    -0.39740338  1.0000000
```

We see high correlation between bathroom number and house size, and between room number and house size. This intuitively makes a lot of sense, bigger houses tend to have more bedrooms and bathrooms. All the other variables are somewhat correlated, most ranging from 0.2 to 0.6, besides the ones mentioned previously. For example, garage size and lot size only have a correlation of 0.204, whereas garage size and room nuber have a correlation of 0.589.

b)

```r
properties_reg <- lm(Y ~ BATH + LOT + SIZE + GAR + ROOM + BED + AGE,
                     data = properties)
summary(properties_reg)
```

```
##
## Call:
## lm(formula = Y ~ BATH + LOT + SIZE + GAR + ROOM + BED + AGE,
##     data = properties)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8792 -1.7515 -0.2857  1.6013  6.1100
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11.62410    5.52343   2.105   0.0515 .
## BATH        10.85551    3.97518   2.731   0.0148 *
## LOT          0.54253    0.46215   1.174   0.2576
## SIZE         3.92170    4.48393   0.875   0.3947
## GAR          2.94143    1.37361   2.141   0.0480 *
## ROOM         2.39374    1.86928   1.281   0.2186
## BED         -4.78958    2.79845  -1.712   0.1063
## AGE         -0.06972    0.05665  -1.231   0.2362
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.13 on 16 degrees of freedom
## Multiple R-squared:  0.8115, Adjusted R-squared:  0.7291
## F-statistic: 9.843 on 7 and 16 DF,  p-value: 8.944e-05
```

We get a fairly high R-squared and adjusted R-squared, showing that the response variables explain the sale price fairly well. However, we see that only `BATH` and `GAR` have statistically significant beta values. `BATH` has a large estimated beta, where as the rest are small, and `BED` and `AGE` having negative betas. This seems right for age (an older house should cost less) but not for bed (more bedrooms should cost more).

c)

```r
properties_reg2 <- lm(Y ~ BATH + GAR, data = properties)
summary(properties_reg2)
```

```
##
## Call:
## lm(formula = Y ~ BATH + GAR, data = properties)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.2211 -3.1169 -0.0322  1.8592 11.3342
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   11.872      4.205   2.824 0.010178 *
## BATH          15.943      3.536   4.509 0.000192 ***
## GAR            3.167      1.408   2.249 0.035372 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.979 on 21 degrees of freedom
## Multiple R-squared:  0.6001, Adjusted R-squared:  0.562
## F-statistic: 15.76 on 2 and 21 DF,  p-value: 6.614e-05
```

```
properties_reg3 <- lm(Y ~ BATH + GAR + AGE, data = properties)
summary(properties_reg3)
```

```
##
## Call:
## lm(formula = Y ~ BATH + GAR + AGE, data = properties)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5769 -2.4129 -0.6249  1.4750  9.1502
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 18.12865    4.34308   4.174 0.000468 ***
## BATH        15.10853    3.11188   4.855  9.6e-05 ***
## GAR          3.17542    1.23329   2.575 0.018086 *
## AGE         -0.14133    0.05202  -2.717 0.013279 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.485 on 20 degrees of freedom
## Multiple R-squared:  0.7079, Adjusted R-squared:  0.6641
## F-statistic: 16.16 on 3 and 20 DF,  p-value: 1.434e-05
```

In both of these regressions, the R-squared and adjusted R-squared values drop, to 0.6001 and 0.562 in `reg2`, and 0.7079 and 0.6641 in `reg3`. We see that `reg3`, which includes `AGE`, does slightly better than `reg2`, showing that age is a valuable predictor. Overall, the drops in R-squared for these models is not too great for the amount of simplification you get from only using 2 or 3 predictors instead of 7.

d)

```
properties_pca <- prcomp(~LOT + SIZE + GAR + ROOM + BED, data = properties, scale = TRUE)
properties_pca$rotation
```

```
##            PC1        PC2        PC3        PC4          PC5
## LOT  0.3096733 -0.7731703 -0.2949459 -0.4222795 -0.202477293
```

```
## SIZE 0.4704060 -0.3395186  0.2899932  0.7611254  0.006113792
## GAR  0.3965764  0.3587489 -0.8130270  0.2250860 -0.048475150
## ROOM 0.5364806  0.1406204  0.1844288 -0.3450071  0.734418524
## BED  0.4875822  0.3721020  0.3659016 -0.2695822 -0.645945167
```

```
#interpret PC1
```

For the most part, PC1 is about the average of the five variables, give and take a few percent.

e)

```
PC1 = predict(properties_pca)[,1]
properties_reg4 <- lm(Y ~ BATH + PC1 + AGE, data = properties)
summary(properties_reg4)
```

```
##
## Call:
## lm(formula = Y ~ BATH + PC1 + AGE, data = properties)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.8810 -2.3283 -0.2991  2.2205  7.7952
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 27.76255    5.09594   5.448 2.48e-05 ***
## BATH        10.86330    3.76480   2.885  0.00915 **
## PC1          1.37620    0.51665   2.664  0.01492 *
## AGE         -0.15503    0.05184  -2.991  0.00723 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.455 on 20 degrees of freedom
## Multiple R-squared:  0.7129, Adjusted R-squared:  0.6699
## F-statistic: 16.56 on 3 and 20 DF,  p-value: 1.209e-05
```

Here, we see that all three of the predictor variables have statistically significant betas. `BATH` still appears to be the strongest predictor with a beta of 10.86, while `PC1` has a smaller beta of 1.38 and `AGE` has a negative beta of -0.16, which is expected since older houses should cost less. The R-squared and adjusted R-squared are relatively high, higher than the other two alternate models `reg2` and `reg3`, though not as high as when every variable was used as a predictor. This shows us that using `PC1` as a predictor does an adequate job for this analysis, as it allows you to reduce the number of predictors but maintain predictive power.