Homework 5

Zach White

9/21/2016

Problem 1

$$\begin{aligned}
P_{X^T} &= (X^TX)(X^TX)^- \\
&= (X^TX)(X^TX)^-(X^TX)(X^TX)^- \\
&= (X^TX)(X^TX)^- \\
&= P_{X^T}
\end{aligned}$$

Since $(X^TX)^-$ is a generalized inverse $(X^TX)(X^TX)^-(X^TX) = (X^TX)$.

Let $w \in C(X^T)$

$$\begin{aligned}
(X^TX)(X^TX)^-w &= X^T(X(X^TX)^-w) \\
&= X^TA \in C(X^T)
\end{aligned}$$

Since since $A = (X(X^TX)^-w) \subset n \times 1$. And thus by defintion, $X^TA \in C(X^T)$

It is clear now that since this condition is met, it is a projection. Since this is a generalized inverse, it doesn't depend on the choice of the inverse. We will now see if it meets the criteria to be considered an orthogonal projection.

$$\begin{aligned}
(P_{X^T})^T &= ((X^TX)(X^TX)^-)^T \\
&= ((X^TX)^-)^T(X^TX)^T \\
&= ((X^TX)^-)^T(X^TX)
\end{aligned}$$

Thus, it is not an orthogonal projection because the symmetry condition doesn't hold.

Problem 2

We know $\lambda = X^Ta$ and $a \in C(X)$.

$$\begin{aligned}
(I - P_{X^T})\lambda &= (I - P_{X^T})X^Ta \\
&= X^Ta - P_{X^T}X^Ta \\
&= X^Ta - X^Ta \\
&= 0
\end{aligned}$$

Problem 3

We know that $X^TX = U\Lambda U^T$ and the Moore Penrose Generalized Inverse is characterized by $A^-_{MP} = U\Lambda^- U^T$.

$$\begin{aligned}
I - P_{X^T} &= I - (X^TX)(X^TX)^- \\
&= I - (U\Lambda U^T)(U\Lambda^- U^T) \\
&= I - U\Lambda\Lambda^- U^T \\
&= UU^T - U\Lambda\Lambda^- U^T \\
&= U(I - \Lambda\Lambda^-)U^T
\end{aligned}$$

Now, $I - \Lambda\Lambda^-$ is a diagonal matrix where all non-zero eigenvalues will be zero, and all the zero eigenvalues will be 1. U and U^T will be a matrix where the columns are the eigenvectors of their corresponding eigenvalues.

Problem 4

We know that $\psi = \lambda^T\beta$ and $\lambda = a^T X$, which means that $\lambda = (\lambda^T)^T = (a^T X)^T = X^T a$. By this we know that $\lambda \in C(X^T)$, and $C(X^T) \subset \mathbb{R}^{p+1}$. No since X is full rank, we know that $C(X^T)$ spans $\mathbb{R}^{p+1}$. Now since these are linearly independent by the definition of rank, it forms a span of $\mathbb{R}^{p+1}$, and since $x_* \in \mathbb{R}^{p+1}$, by Gauss-Markov theorem, we have a unique unbiased linear estimator. Thus, this also also for $x_* \in \mathbb{R}^{p+1}$.

Problem 5

Optional

Problem 6

```
data(Prostate)

Prostate$g6 = Prostate$gleason == 6
Prostate$g7 = Prostate$gleason == 7
Prostate$g8 = Prostate$gleason == 8
Prostate$g9 = Prostate$gleason == 9

Prostate$g6 +Prostate$g7 +Prostate$g8 + Prostate$g9
```

```
##  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [36] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [71] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

When we add the dummy variables, we get the intercept, and thus these are clearly linearly related to the intercept.

Problem 7

```
lpsa6789 = lm(lpsa~g6+g7+g8+g9, data = Prostate)
lpsa6987 = lm(lpsa~g6+g9+g8+g7, data = Prostate)
lpsa7968 = lm(lpsa~g7+g9+g6+g8, data = Prostate)
lpsa9876 = lm(lpsa~g9+g8+g7+g6, data = Prostate)
coefficients(lpsa6789)
```

```
## (Intercept)      g6TRUE      g7TRUE      g8TRUE      g9TRUE
##  2.87317976 -1.13480987  0.03819743 -0.71562044          NA
```

```
coefficients(lpsa6987)
```

```
## (Intercept)      g6TRUE      g9TRUE      g8TRUE      g7TRUE
##  2.91137719 -1.17300730 -0.03819743 -0.75381787          NA
```

```
coefficients(lpsa7968)
```

```
## (Intercept)      g7TRUE      g9TRUE      g6TRUE      g8TRUE
##   2.1575593   0.7538179   0.7156204  -0.4191894          NA
```

```r
coefficients(lpsa9876)
```

```
## (Intercept)      g9TRUE      g8TRUE      g7TRUE      g6TRUE
##   1.7383699   1.1348099   0.4191894   1.1730073          NA
```

```r
lpsa6789.noint = lm(lpsa~ -1 +g6+g7+g8+g9, data = Prostate)
lpsa6987.noint = lm(lpsa~-1+g6+g9+g8+g7, data = Prostate)
lpsa7968.noint = lm(lpsa~-1+g7+g9+g6+g8, data = Prostate)
lpsa9876.noint = lm(lpsa~-1+g9+g8+g7+g6, data = Prostate)
coefs.1.not = coefficients(lpsa6789.noint)
coefs.1.not
```

```
##     g6FALSE      g6TRUE      g7TRUE      g8TRUE      g9TRUE
##  2.87317976  1.73836989  0.03819743 -0.71562044          NA
```

```r
coefficients(lpsa6987.noint)
```

```
##     g6FALSE      g6TRUE      g9TRUE      g8TRUE      g7TRUE
##  2.91137719  1.73836989 -0.03819743 -0.75381787          NA
```

```r
coefficients(lpsa7968.noint)
```

```
##    g7FALSE     g7TRUE     g9TRUE     g6TRUE     g8TRUE
##  2.1575593  2.9113772  0.7156204 -0.4191894         NA
```

```r
coefs.2.not = coefficients(lpsa9876.noint)
coefs.2.not
```

```
##   g9FALSE    g9TRUE    g8TRUE    g7TRUE    g6TRUE
## 1.7383699 2.8731798 0.4191894 1.1730073        NA
```

When we change the order, of the variables, the intercepts adopt the value of the variable ommitted. For example, in the first model shown lpsa6789, the last variable is g9, and it is ommitted. The intercept takes on the value 2.873, which is nearly equivalent to the model where g9 is the first variable and the intercept is ommitted. When we change the order, the last variable of the order is ommitted in the model because by that time, we have enough variables to form a linear combination to get that last model.

When we force the intercept to be zero, the first coefficient becomes the value of the variable ommitted by the model, it shows as g6FALSE or something of the sort, but when we compare the first model, where the order is g6,g7,g8,g9, the coefficients are 2.8731798, 1.7383699, 0.0381974, -0.7156204, NA. The last model, the order is g9,g8,g7,g6, and the coefficients are 1.7383699, 2.8731798, 0.4191894, 1.1730073, NA. Now, in this case, the first values are just switched. So it's as if the first one, the value takes the value of g9 and in the second model, the first value takes the value of g6TRUE from the previous model.

Problem 8

```r
lpsa.gleason = lm(lpsa~as.factor(gleason), data = Prostate)
coefficients(lpsa.gleason)
```

```
##         (Intercept) as.factor(gleason)7 as.factor(gleason)8
##           1.7383699           1.1730073           0.4191894
## as.factor(gleason)9
##           1.1348099
```

```r
head(model.matrix(lpsa.gleason))
```

```
##   (Intercept) as.factor(gleason)7 as.factor(gleason)8 as.factor(gleason)9
## 1           1                   0                   0                   0
```

```
## 2           1                0                0                0
## 3           1                1                0                0
## 4           1                0                0                0
## 5           1                0                0                0
## 6           1                0                0                0
```

```r
exp(coefficients(lpsa.gleason))
```

```
##         (Intercept) as.factor(gleason)7 as.factor(gleason)8
##            5.688064            3.231697            1.520728
## as.factor(gleason)9
##            3.110582
```

```r
lpsa.noint = lm(lpsa~-1+g7+g8+g9+g6, data = Prostate)
coefficients(lpsa.noint)
```

```
##   g7FALSE    g7TRUE    g8TRUE    g9TRUE    g6TRUE
## 1.7383699 2.9113772 0.4191894 1.1348099        NA
```

The equivalent model is when we don't impose an intercept of 0, and we order the models as g9 + g8 + g7 + g6. Thus we can interpret the intercept as a baseline of prostate specific antigen as 5.688, and then each other coefficient would be an increase from there if they get a different score. For example, while holding everything else constant, if someone were to increase to a level 7 from 6, then the prostate specific antigent would increase 1.173 on average.

Problem 9

With the model with the intercept and the all the dummies, show it is not estimable using stuff from class.

```r
lpsa6789 = lm(lpsa~g6+g7+g8+g9, data = Prostate)
X = model.matrix(lpsa6789)
xtx = t(X) %*% X
xtx.inv = ginv(xtx)
tot.x = xtx %*% xtx.inv

(diag(5) - tot.x) %*% diag(5)
```

```
##             [,1] [,2] [,3] [,4] [,5]
## (Intercept)  0.2 -0.2 -0.2 -0.2 -0.2
## g6TRUE      -0.2  0.2  0.2  0.2  0.2
## g7TRUE      -0.2  0.2  0.2  0.2  0.2
## g8TRUE      -0.2  0.2  0.2  0.2  0.2
## g9TRUE      -0.2  0.2  0.2  0.2  0.2
```

```r
lm.dummy.age = lm(lpsa~g6+g7+g8+g9 + age, data = Prostate)
X = model.matrix(lm.dummy.age)
xtx = t(X) %*% X
xtx.inv = ginv(xtx)
tot.x.1 = xtx %*% xtx.inv
(diag(6) - tot.x.1) %*% diag(6)
```

```
##                      [,1]          [,2]          [,3]          [,4]
## (Intercept)  2.000000e-01 -2.000000e-01 -2.000000e-01 -2.000000e-01
## g6TRUE      -2.000000e-01  2.000000e-01  2.000000e-01  2.000000e-01
## g7TRUE      -2.000000e-01  2.000000e-01  2.000000e-01  2.000000e-01
## g8TRUE      -2.000000e-01  2.000000e-01  2.000000e-01  2.000000e-01
## g9TRUE      -2.000000e-01  2.000000e-01  2.000000e-01  2.000000e-01
```

```
## age          3.577852e-12  1.024365e-11  4.526605e-12 -1.157698e-11
##                      [,5]          [,6]
## (Intercept) -2.000000e-01 -2.615681e-14
## g6TRUE       2.000000e-01 -6.903115e-15
## g7TRUE       2.000000e-01 -7.174599e-15
## g8TRUE       2.000000e-01 -5.917359e-15
## g9TRUE       2.000000e-01 -6.054185e-15
## age          7.347118e-14 -1.525446e-13
```

Thus it is clear that none of the ones in the first matrix are estimable. But when we add age into the model, something we assume to be estimable, those corresponding vectors will equal 0.

Problem 10

Optional