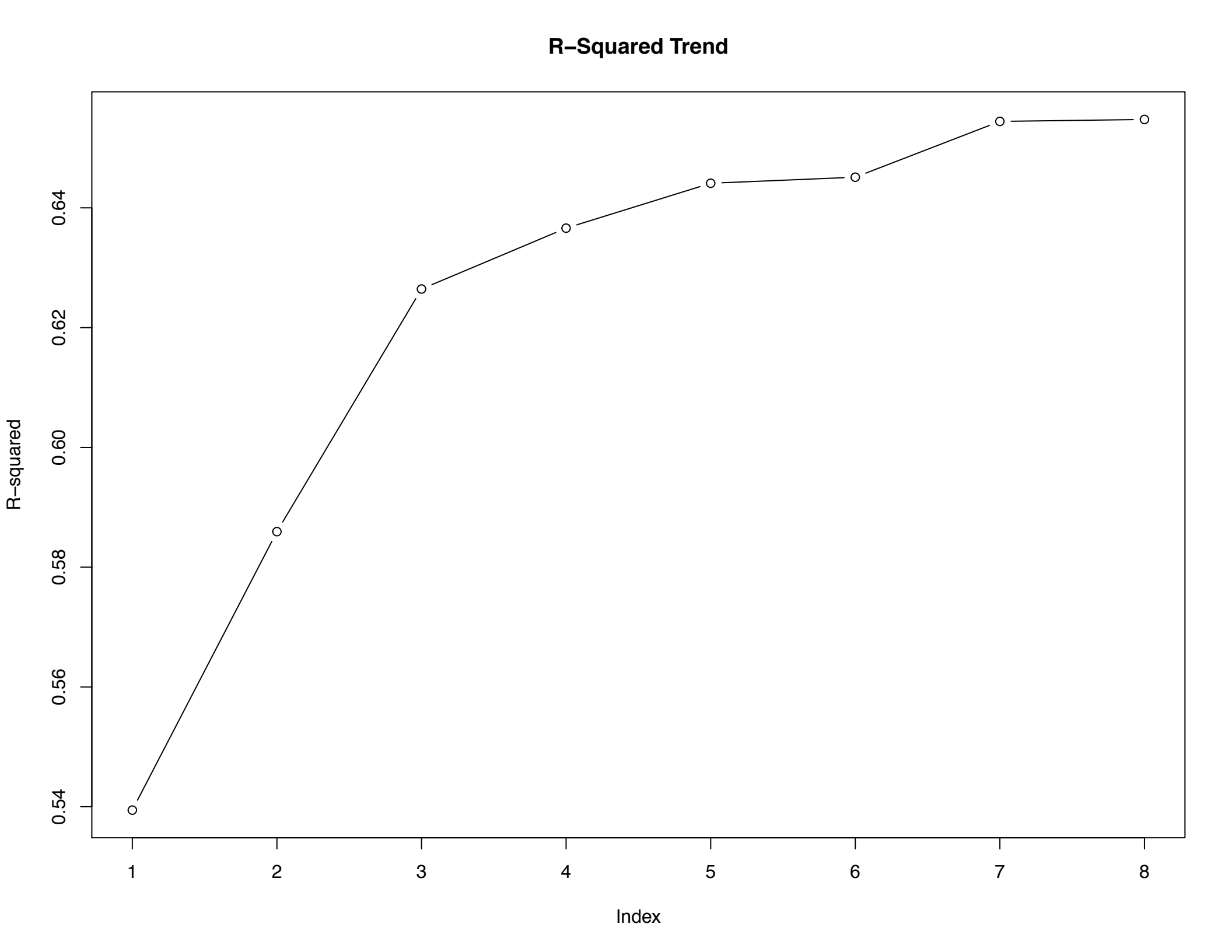
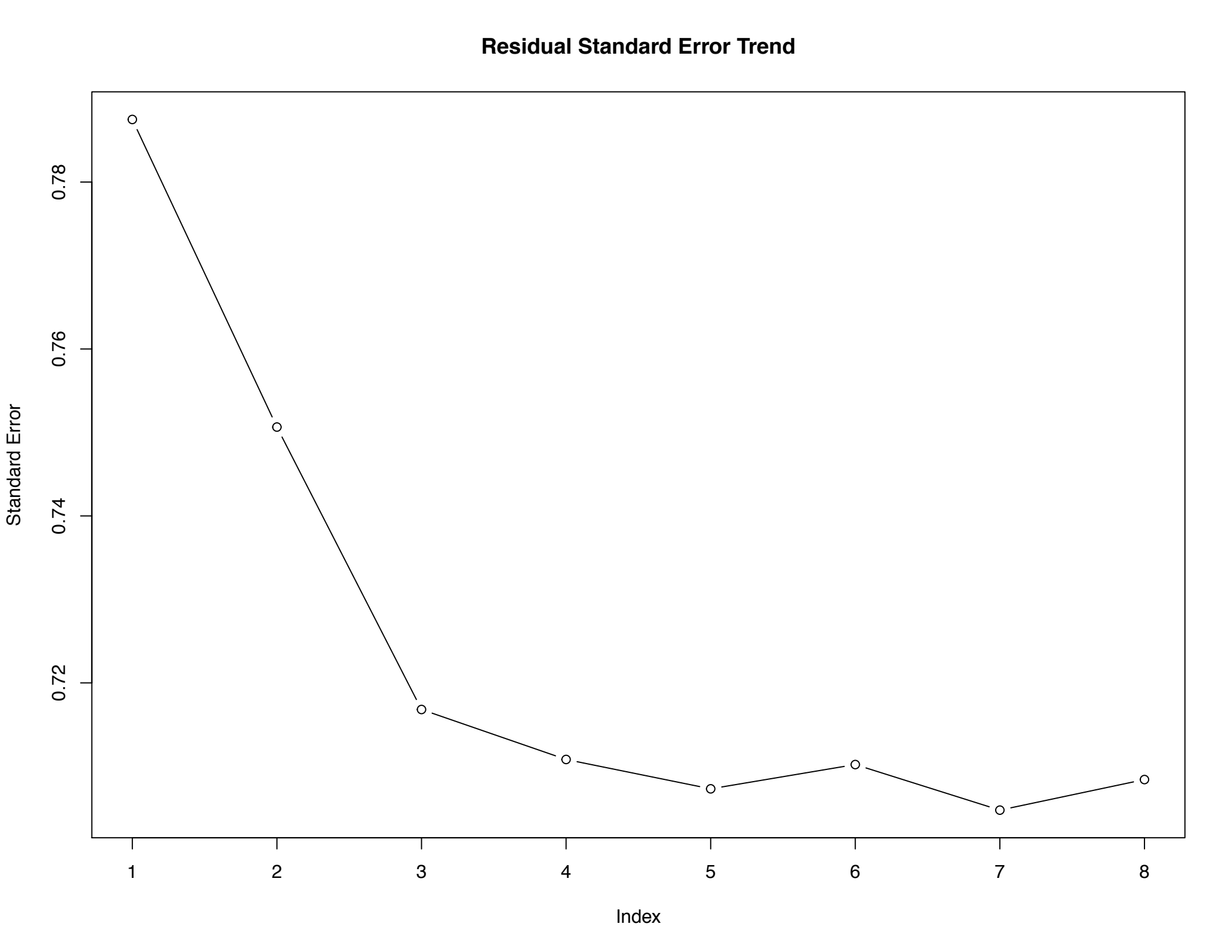
```
# title: "MSDS596 - HW1"
# author: "Diego Sarachaga"
# date: "09/18/2018"


#Problem 1
#Part a
  # λ is eigenvalue of A => Ax = λx
  # A is idemponent => A²x = λx => A(Ax) = λx
  # x is eigenvector of A => A(λx) = λx => λ(Ax) = λx
  # x is eigenvector of A => λ(λx) = λx => λ²x = λx
  # λ²x - λx = 0 => x(λ² - λ) = 0
  # => λ = 0 or λ = 1


#Part b
  # *1
  # Ax = λx => rank(A) = ∑ λi, λi != 0   # (sum from i to n of λ sub i)

  # *2
  # Spectral decomposition A = QΛQ'
  # tr(A) = tr(QΛQ') = tr(QQ'Λ)  #trace properties
  # QQ' = I => tr(QQ'Λ) = tr(Λ) = ∑ λi

  # *3
  # using part a) λ = 0 or λ = 1

  # because of *1, *2 and *3 => tr(A) = rank(A)

# Part c
  # A is idemponent => (I-A)² = I² - 2IA + A² = I - 2A + A = I-A => I-A is
idemponent

  # using part b
  # rank(I-A) = tr(I-A)
  # tr(I-A) = tr(I) - tr(A)  # trace properties
  # tr(I) - tr(A) = n - rank(A)
  # => rank(I-A) = n - rank(A) => rank(A) + rank(I-A) = n



# Problem 2
# (I-P)² = I² - 2IP + P² = I - 2P + P = I-P => I-P is idemponent
# I-P is also symmetric
# so I-P = In - 1/n(1n)(1n') is a projection matrix
# and P = 1/n(1n)(1n')

# A projection matrix is a square matrix that does a linear transformation of
a vector onto a subspace


#install.packages("faraway")
library(faraway)

#Problem 5
data(teengamb)
head(teengamb)
```

```
#using linear model function
aux <- lm(gamble ~ sex+status+income+verbal)

#part a
print("Part a")
#getting summary of the model
print(summary(aux))

#part b
print("Part b")
#getting the r-square value from the summary
print(summary(aux)$r.squared)

#part c
print("Part c")
#max:
print("max:")
#getting the max value of all model's residuals
print(which.max(aux$residuals))
print("min:")
#min:
#getting the min value of all model's residuals
print(which.min(aux$residuals))

#part d
print("Part d")
print("Mean:")
#mean
#getting the mean value of all model's residuals
print(mean(aux$residuals))
print("Median:")
#median
#getting the median value of all model's residuals
print(median(aux$residuals))

#part e
print("Part e")
#getting the correlation between the residuals and the fitted values of the
model
print(cor(aux$residuals,aux$fitted.values))

#part f
print("Part f")
#getting the correlation between the residuals and the income
print(cor(aux$residuals,income))

#part g
print("Part g")
print("Based on the summary, the fitted model can be explicitly written as:
gamble = 22.55565 - 22.11833 × sex + 0.05223 × status + 4.96198 × income -
2.95949 × verbal")

# from
https://www.rdocumentation.org/packages/faraway/versions/1.0.7/topics/teengamb

# sex is 0 for man and 1 for female

print("If all the predictors except sex continue to be constant, the
difference in predicted
```

```
expenditure on gambling between male and female will be equal to the
regression
coefficient of sex (-22.11833). So if sex changes from male (0) to female
(1),
the value of gamble decreases by 22.11833.")


#Problem 6
data(prostate)
head(prostate)

# creating vectors to save r-square and standard error values from different
models
r_squared_trend <- c()
se_trend <- c()

#part a
# creating the model with lpsa as the response and lcavol as the predictor
prostate.lm1=lm(lpsa~lcavol, data=prostate);
lm1.rsquared <-summary(prostate.lm1)$r.squared
lm1.sigma <-summary(prostate.lm1)$sigma

#adding r-square and standard error values to the vectors
r_squared_trend <- c(r_squared_trend, lm1.rsquared)
se_trend <- c(se_trend, lm1.sigma)


#part b
#adding lweight
prostate.lm2=lm(lpsa~lcavol+lweight, data=prostate);
#getting the r-square value
lm2.rsquared <-summary(prostate.lm2)$r.squared
#getting the standard error value
lm2.sigma <-summary(prostate.lm2)$sigma

#adding r-square and standard error values to the vectors
r_squared_trend <- c(r_squared_trend, lm2.rsquared)
se_trend <- c(se_trend, lm2.sigma)

#adding svi
prostate.lm3=lm(lpsa~lcavol+lweight+svi, data=prostate);
lm3.rsquared <-summary(prostate.lm3)$r.squared
lm3.sigma <-summary(prostate.lm3)$sigma

#adding r-square and standard error values to the vectors
r_squared_trend <- c(r_squared_trend, lm3.rsquared)
se_trend <- c(se_trend, lm3.sigma)

#adding lbph
prostate.lm4=lm(lpsa~lcavol+lweight+svi+lbph, data=prostate);
lm4.rsquared <-summary(prostate.lm4)$r.squared
lm4.sigma <-summary(prostate.lm4)$sigma

#adding r-square and standard error values to the vectors
r_squared_trend <- c(r_squared_trend, lm4.rsquared)
se_trend <- c(se_trend, lm4.sigma)

#adding age
prostate.lm5=lm(lpsa~lcavol+lweight+svi+lbph+age, data=prostate);
lm5.rsquared <-summary(prostate.lm5)$r.squared
```

```
lm5.sigma <-summary(prostate.lm5)$sigma

#adding r-square and standard error values to the vectors
r_squared_trend <- c(r_squared_trend, lm5.rsquared)
se_trend <- c(se_trend, lm5.sigma)

#adding lcp
prostate.lm6=lm(lpsa~lcavol+lweight+svi+lbph+age+lcp, data=prostate);
lm6.rsquared <-summary(prostate.lm6)$r.squared
lm6.sigma <-summary(prostate.lm6)$sigma

#adding r-square and standard error values to the vectors
r_squared_trend <- c(r_squared_trend, lm6.rsquared)
se_trend <- c(se_trend, lm6.sigma)

#adding pgg45
prostate.lm7=lm(lpsa~lcavol+lweight+svi+lbph+age+lcp+pgg45, data=prostate);
lm7.rsquared <-summary(prostate.lm7)$r.squared
lm7.sigma <-summary(prostate.lm7)$sigma

#adding r-square and standard error values to the vectors
r_squared_trend <- c(r_squared_trend, lm7.rsquared)
se_trend <- c(se_trend, lm7.sigma)

#adding gleason
prostate.lm8=lm(lpsa~lcavol+lweight+svi+lbph+age+lcp+pgg45+gleason,
data=prostate);
lm8.rsquared <-summary(prostate.lm8)$r.squared
lm8.sigma <-summary(prostate.lm8)$sigma

#adding r-square and standard error values to the vectors
r_squared_trend <- c(r_squared_trend, lm8.rsquared)
se_trend <- c(se_trend, lm8.sigma)

#Plot R-Square
plot(r_squared_trend, type="b", main="R-Squared Trend", ylab="R-squared")
#R-Squared: when adding more variables to model, it is closer to 1. So we can
conclude the more
#variables we include in the model, it better fits.

#Plot Standard Error
plot(se_trend, type="b", main="Residual Standard Error Trend", ylab="Standard
Error")
#Residual Standard Error: Contrary to R-Squared, when adding more variables
to the model, it is closer
#to 0, which is good, because it means that as we aggregate variable, the
model fits better.
```


R–Squared Trend
R–squared
0.54
0.56
0.58
0.60
0.62
0.64
1
2
3
4
5
6
7
8
Index


Residual Standard Error Trend
Standard Error
0.72
0.74
0.76
0.78
1
2
3
4
5
6
7
8
Index