ANLY 500 Final Project

Data Cleaning

- Unzip MERGED2014_15_PP.csv.zip, it is from College Scorecard webiste, not Kaggle. Since Kaggle do not have latest data.
- `mrc_table10` is data from Mobility Report Card data, they give each school a tier. Looks more useful than payscale school type. For detail, please see Codebook-MRC-Table-6.pdf.

Read Data

```
payscale_college_type = read.csv('salaries-by-college-type.csv')
payscale_college_type = payscale_college_type[, c(1,2)]
payscale_region = read.csv('salaries-by-region.csv')
payscale_region = payscale_region[, c(1,2)]

mrc_table10 = read.csv('mrc_table10.csv')
mrc_table10 = mrc_table10[, c(1, 12)]

# Treat "NULL" and "PrivacySuppressed" as NA when read
college_scorecard = read.csv('MERGED2014_15_PP.csv', na=c("NULL", "PrivacySuppressed"))
college_scorecard = college_scorecard[, c(3, 4,6,17,377,1639,1640,1642,1643,1645,1655,1656,1657,1661,16
```

Merge Data

Process School Name

Normalize all school name for merging.

```
process_school_name = function(data) {
  data = sub(" \\(.*\\)", "", data)
  data = sub(" - ", "-", data)
  data = sub(", ", "-", data)
  data = sub("\\.", "", data)
  data = sub(" & ", " and ", data)
  data = sub("&", " and ", data)
  data = sub("St ", "Saint ", data)
  data
}

payscale_college_type$School.Name = process_school_name(payscale_college_type$School.Name)
nrow(payscale_college_type)
```

```
## [1] 269
```

```
payscale_region$School.Name = process_school_name(payscale_region$School.Name)
nrow(payscale_region)
```

```
## [1] 320
```

```r
college_scorecard$INSTNM = process_school_name(college_scorecard$INSTNM)
nrow(college_scorecard)
```

```
## [1] 7703
```

Merge Payscale data

```r
payscale = merge(payscale_college_type, payscale_region, by="School.Name", all = FALSE)

payscale %>% group_by(School.Name) %>% filter(n() > 1)

payscale_party = payscale %>%
  filter(School.Type == "Party")

payscale = payscale %>%
  filter(School.Type != "Party") %>%
  mutate(Is.Party = School.Name %in% payscale_party$School.Name)

nrow(payscale)
```

All the duplicate rows in Payscale data is because it duplicates all Party Schools. So we split whether or not is a party school into a separate column.

Merge with College Scorecard

```r
scorecard_payscale = merge(payscale, college_scorecard, by.y = "INSTNM", by.x = "School.Name")

scorecard_payscale %>% group_by(School.Name) %>% filter(n() > 1)

scorecard_payscale = scorecard_payscale %>%
  filter( !((School.Name == 'Union College' & STABBR != 'NY') |
            (School.Name == 'Wentworth Institute of Technology' & is.na(COSTT4_A))))

nrow(scorecard_payscale)
```

Two schools are duplicate after merged with College Scorecard data. After some search, we only keep Union College in New York because that is the only one in northwest. And we only keep one Wentworth Institute of Technology because the others do not have data.

Merge with MRC

```r
data = merge(scorecard_payscale, mrc_table10, by.x = "OPEID6", by.y = "super_opeid")

data %>% group_by(School.Name) %>% filter(n() > 1)

nrow(data)
```

No duplicate in this step

Other processing

```
names(data)[names(data) == 'tier_name'] <- 'Tier'
earning_colnames = c("MN_EARN_WNE_P6", "MD_EARN_WNE_P6", "PCT25_EARN_WNE_P6", "PCT75_EARN_WNE_P6", "SD_I
cost_colnames = c("COSTT4_A")
attr_colnames = c("School.Name", "School.Type", "Region", "Is.Party",
"STABBR", "CONTROL", "Tier")

data$CONTROL = factor(data$CONTROL, levels = c(1,2), labels = c("Public", "Private nonprofit"))
```

We reorder the column for easy inspection. And convert `CONTROL` into factor.

Accuracy and Outlier

Accurary & Missing value

```
summary(data)

percentmiss <- function(x){length(x[is.na(x)])/length(x)*100}

# process column first will get more records left
missing_col = apply(data, 2, percentmiss)
missing_col
delete <- which(missing_col > 5)
replace_col = data[,-delete]
dont_col = data[,delete]

missing_row = apply(replace_col, 1, percentmiss)
missing_row[missing_row > 5]
replace_row = subset(replace_col, missing_row <= 5)
dont_row = subset(replace_col, missing_row > 5)

# change to "cart" to avoid error, increase iteration to get reliable result
temp_no_miss = mice(replace_row, maxit=100, method='cart', seed=500)
no_miss = mice::complete(temp_no_miss,1)

# combine data back
all_rows = rbind(dont_row, no_miss)
all_col = cbind(dont_col, all_rows)
```

There is no accuracy problem in the data. We use mice to complete the missing value for data meet 5% rule.

Outlier

```
# pass tolerance to prevent mahalanobis think it is singular matrix
attr_index = which(colnames(no_miss) %in% attr_colnames)
mahal <- mahalanobis(no_miss[-c(attr_index)],
                     colMeans(no_miss[-c(attr_index)],na.rm=TRUE),
                     cov(no_miss[-c(attr_index)], use = "pairwise.complete.obs"),
                     tol=1e-30)
```

```r
cutoff = qchisq(1-.001,ncol(no_miss[-c(attr_index)]))
print(cutoff)
```

```
## [1] 43.8202
```

```r
summary(mahal < cutoff)
```

```
##    Mode   FALSE    TRUE
## logical      15     128
```

```r
noout = subset(no_miss, mahal < cutoff)

no_miss[mahal >= cutoff, c("School.Name", "COSTT4_A", "MN_EARN_WNE_P6", "MD_EARN_WNE_P6", "MN_EARN_WNE_I
```

```
##                                        School.Name COSTT4_A MN_EARN_WNE_P6
## 16                                  Pomona College    59730          51200
## 26                         Colorado School of Mines    30154          74700
## 30                                 Yale University    61620          67800
## 55                                 Bowdoin College    59900          57100
## 56                                   Colby College    59110          50200
## 57                                 Amherst College    61544          61600
## 59           Massachusetts Institute of Technology    59020          99600
## 61                                Williams College    61850          51400
## 76                            Princeton University    57400          73600
## 78  New Mexico Institute of Mining and Technology    18762          49000
## 105                                   Reed College    59595          34800
## 108                     Carnegie Mellon University    61990          84000
## 115                     University of Pennsylvania    61800          91200
## 135                   Washington and Lee University    58575          58900
## 143                Florida International University    18784          41500
##     MD_EARN_WNE_P6 MN_EARN_WNE_P10 MD_EARN_WNE_P10
## 16           41100           77300           58100
## 26           69200           95600           84900
## 30           56600          124400           83200
## 55           44600           83300           65500
## 56           42700           71000           58100
## 57           44100           83300           65000
## 59           82200          153600          104700
## 61           42600           89800           59000
## 76           60800          116300           74700
## 78           43500           58500           50000
## 105          30400           52700           42200
## 108          69800          103000           83600
## 115          71600          131600           85900
## 135          49900           93300           76100
## 143          38700           52000           46300
```

```r
data = all_col
```

The outliers are more than 5% of data, so we keep them.

ROI Analysis

```r
## Break-even point calculation
data$BE_MN_P6 = data$COSTT4_A * 4 / data$MN_EARN_WNE_P6
```

```r
data$BE_MD_P6 = data$COSTT4_A * 4 / data$MD_EARN_WNE_P6
data$BE_MN_P10 = data$COSTT4_A * 4 / data$MN_EARN_WNE_P10
data$BE_MD_P10 = data$COSTT4_A * 4 / data$MD_EARN_WNE_P10

## ROI USING EARNINGS of beigining salary
data$ROI_MN_P6 = (data$MN_EARN_WNE_P6 - data$COSTT4_A) / data$COSTT4_A * 100
data$ROI_MD_P6 = (data$MD_EARN_WNE_P6 - data$COSTT4_A) / data$COSTT4_A * 100

## ROI USING MEAN EARNINGS of 6-year salary
data$ROI_MN_P10 = (data$MN_EARN_WNE_P10 - data$COSTT4_A) / data$COSTT4_A * 100
data$ROI_MD_P10 = (data$MD_EARN_WNE_P10 - data$COSTT4_A) / data$COSTT4_A * 100
roi_colnames = c("BE_MN_P6", "BE_MD_P6", "BE_MN_P10", "BE_MD_P10", "ROI_MN_P6", "ROI_MD_P6", "ROI_MN_P1
```

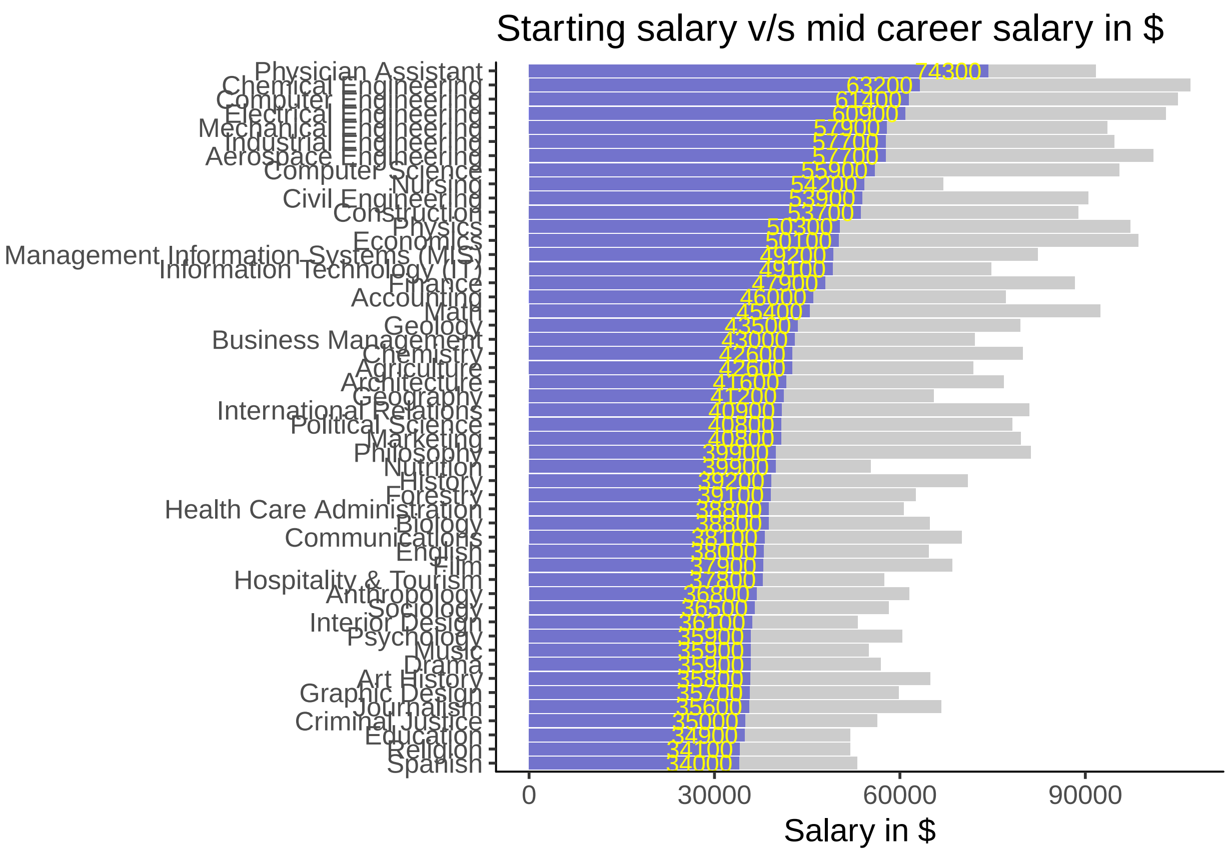
Data Output

```r
data_gender_p6 = melt(data[c("School.Name", "MN_EARN_WNE_MALE0_P6", "MN_EARN_WNE_MALE1_P6")], id=c("Sch
levels(data_gender_p6$Gender) = c("Male", "Female")
colnames(data_gender_p6)[3] = "MN_EARN_WNE_GENDER_P6"

data_gender_p10 = melt(data[c("School.Name", "MN_EARN_WNE_MALE0_P10", "MN_EARN_WNE_MALE1_P10")], id=c("
levels(data_gender_p10$Gender) = c("Male", "Female")
colnames(data_gender_p10)[3] = "MN_EARN_WNE_GENDER_P10"

data_inc_p6 = melt(data[c("School.Name", "MN_EARN_WNE_INC1_P6", "MN_EARN_WNE_INC2_P6", "MN_EARN_WNE_INC
colnames(data_inc_p6)[3] = "MN_EARN_WNE_INC_P6"
levels(data_inc_p6$Family.Income.Tercile) = c("Lowest Income Tercile", "Middle Income Tercile", "Highes

data_inc_p10 = melt(data[c("School.Name", "MN_EARN_WNE_INC1_P10", "MN_EARN_WNE_INC2_P10", "MN_EARN_WNE_
colnames(data_inc_p10)[3] = "MN_EARN_WNE_INC_P10"
levels(data_inc_p10$Family.Income.Tercile) = c("Lowest Income Tercile", "Middle Income Tercile", "Highe

data = data[c(attr_colnames, cost_colnames, earning_colnames, roi_colnames)]
# write_csv(data, "data_cleaned.csv")
# write_csv(data_gender_p6, "data_gender_p6.csv")
# write_csv(data_gender_p10, "data_gender_p10.csv")
# write_csv(data_inc_p6, "data_inc_p6.csv")
# write_csv(data_inc_p10, "data_inc_p10.csv")
```

Algorithm and Models

```r
##Read the 4th dataset
salary_degree <-
  read_csv(
    'degrees-that-pay-back.csv',
    col_names = c(
      "major",
      "start_med_slry",
      "mid_car_slry",
      "percent_chng",
      "mid_car_10th",
```

```r
      "mid_car_25th",
      "mid_car_75th",
      "mid_car_90th"
    ),
    col_types = "cnndnnnn", # specify column types to coerce '$' to numeric
    skip = 1 # names specified, skip header
  )

#####
cleanup<-theme(panel.grid.major = element_blank(),
               panel.grid.minor = element_blank(),
               panel.background = element_blank(),
               axis.line.x = element_line(color = 'black'),
               axis.line.y = element_line(color = 'black'),
               legend.key = element_rect(fill = 'white'),
               text = element_text(size = 12))
##  Plot degree v/s starting salary
p1 <- ggplot(salary_degree, aes(x = reorder(major, start_med_slry), start_med_slry)) +
  geom_col(fill = "blue", alpha = 0.5) +
  geom_col(aes(x = reorder(major, mid_car_slry), mid_car_slry), alpha = 0.3) +
  geom_text(aes(label = (start_med_slry)), size = 3, hjust = 1.1, col="yellow")+
   xlab(NULL) +
  ylab("Salary in $")+
  coord_flip() +
  ggtitle("Starting salary v/s mid career salary in $")+
  cleanup
p1
```




```r
## Top 10 school

accent_colors_edit <- brewer.pal(n = 5, "Pastel1")[c(1:3, 5)]   # keep colors consistent for plot w/o 'p

top10_colleges <- data %>%
  select(School.Name, School.Type, MD_EARN_WNE_P8) %>%
  arrange(desc(MD_EARN_WNE_P8)) %>%
  top_n(10)
```

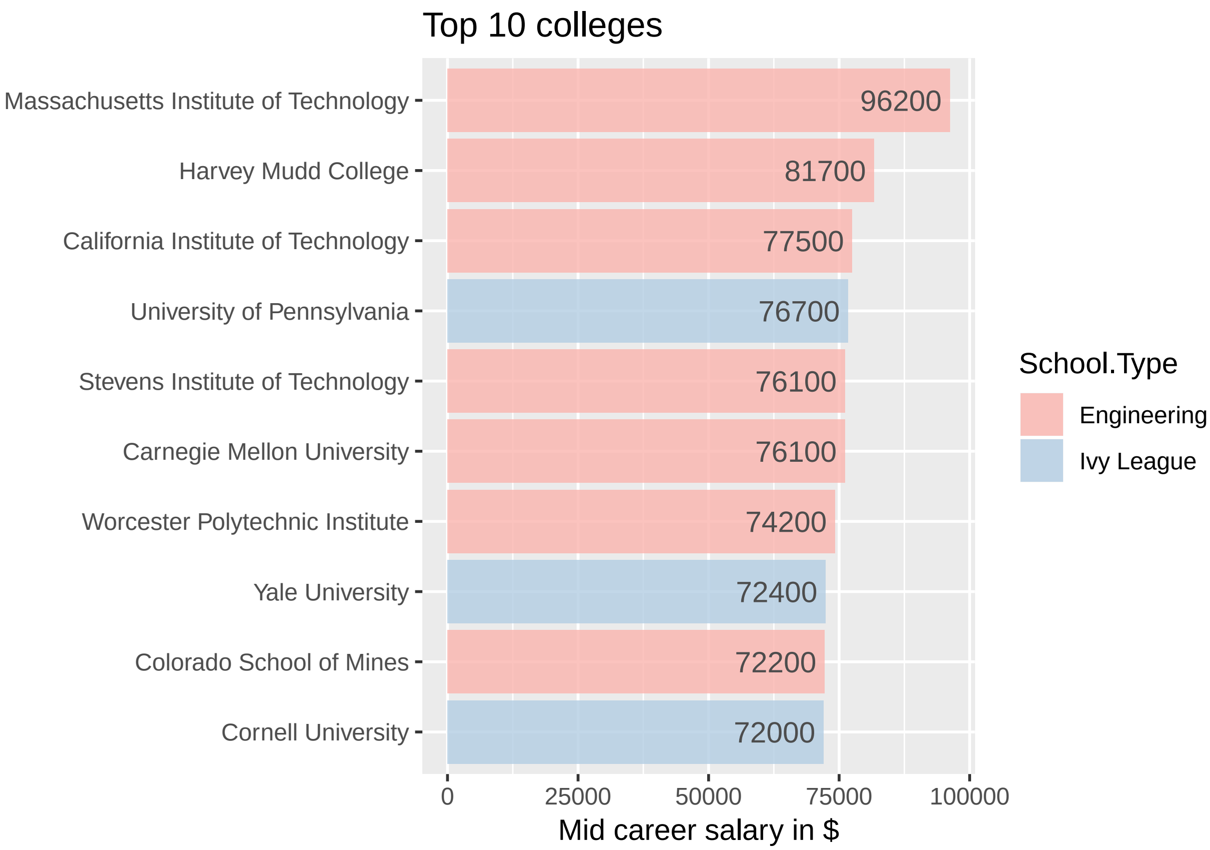
```
## Selecting by MD_EARN_WNE_P8
```

```r
top10_colleges
```

```
##                              School.Name School.Type MD_EARN_WNE_P8
## 1  Massachusetts Institute of Technology Engineering          96200
## 2                     Harvey Mudd College Engineering          81700
## 3     California Institute of Technology Engineering          77500
## 4              University of Pennsylvania  Ivy League          76700
## 5         Stevens Institute of Technology Engineering          76100
## 6              Carnegie Mellon University Engineering          76100
## 7         Worcester Polytechnic Institute Engineering          74200
## 8                         Yale University  Ivy League          72400
## 9               Colorado School of Mines Engineering          72200
## 10                     Cornell University  Ivy League          72000
```

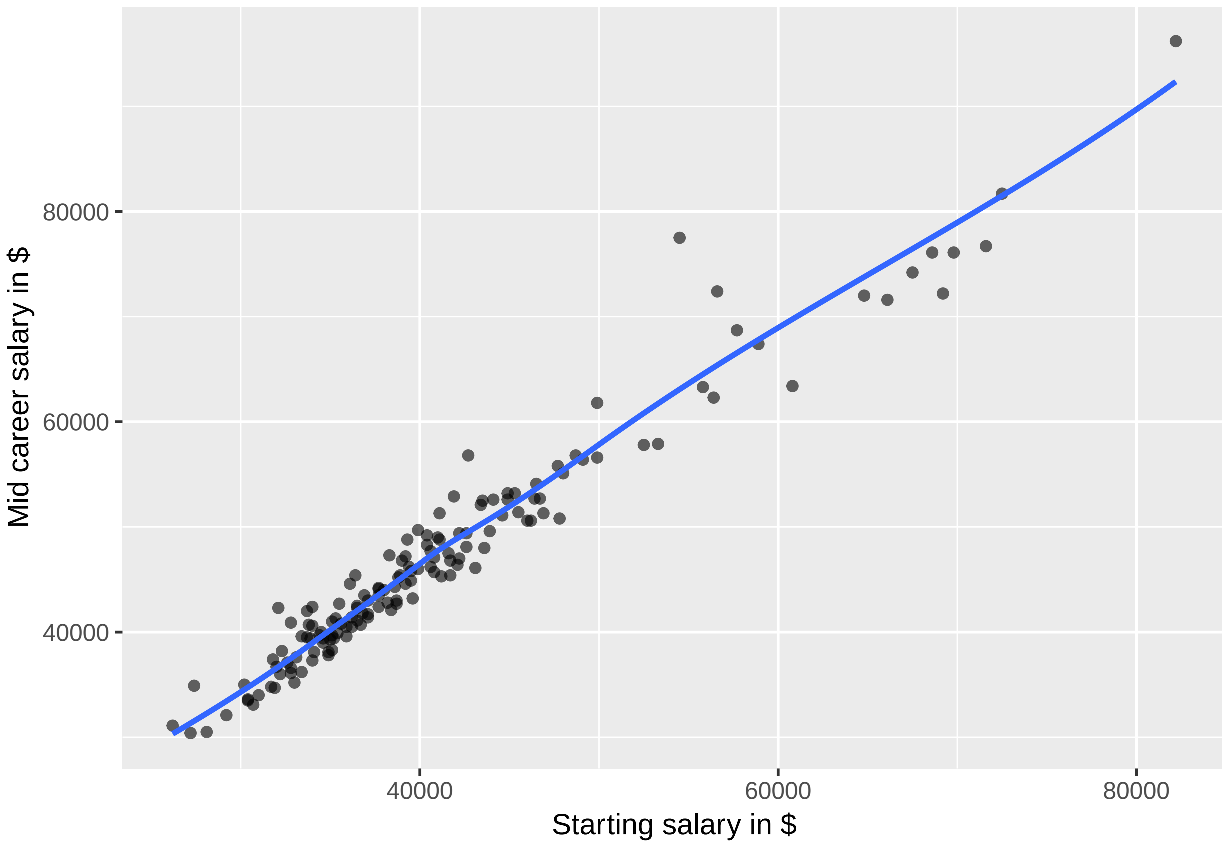
```r
ggplot(top10_colleges, aes(reorder(School.Name, MD_EARN_WNE_P8), MD_EARN_WNE_P8, fill = School.Type)) +
  geom_col(alpha = 0.8) +
  scale_fill_manual(values = accent_colors_edit) +
```

```r
  geom_text(aes(label =(MD_EARN_WNE_P8)), hjust = 1.1, color = 'gray30') +
   xlab(NULL) +
  ggtitle("Top 10 colleges")+
  ylab("Mid career salary in $")+
   coord_flip()
```

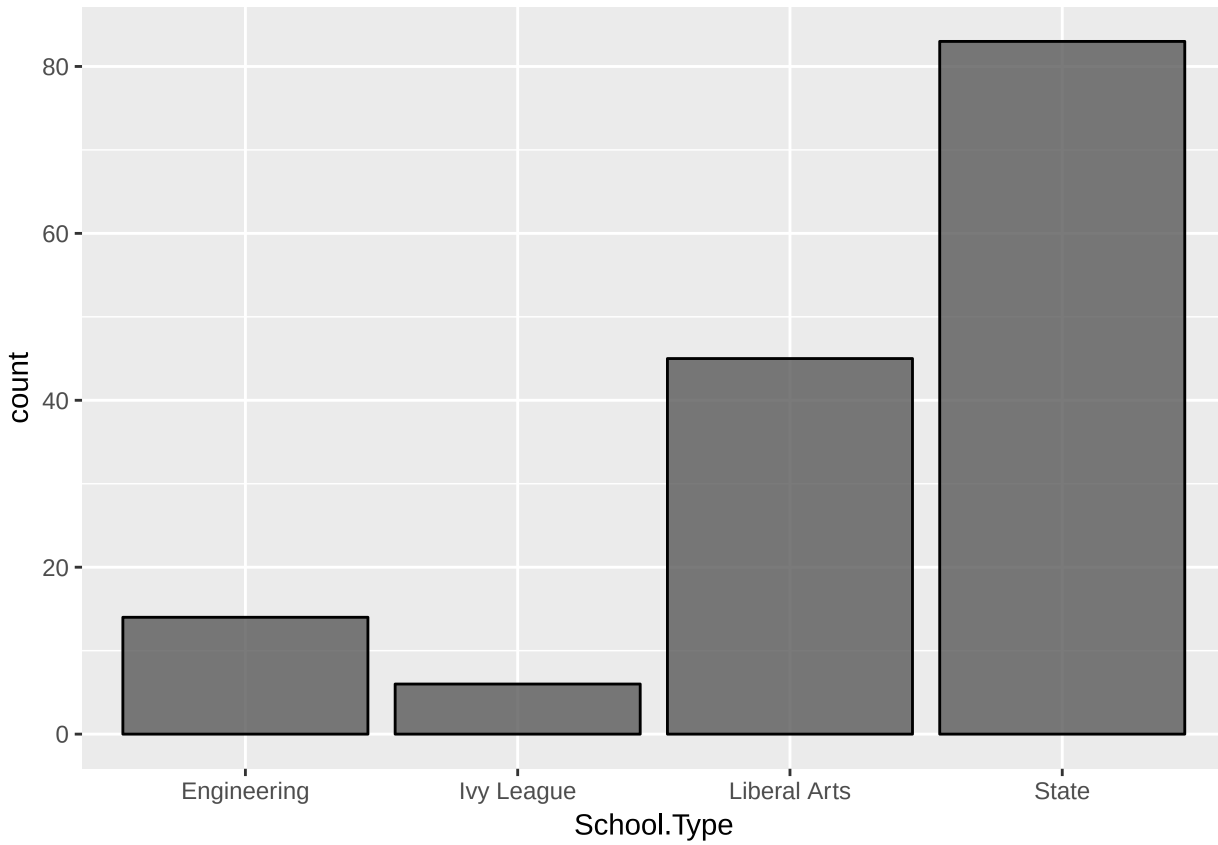



```r
## Any correlation b/w starting salary and mid career salary

ggplot(data, aes(MD_EARN_WNE_P6, MD_EARN_WNE_P8)) +
  geom_point(alpha = 0.6) +
  geom_smooth(se = F) +
  xlab("Starting salary in $")+
  ylab("Mid career salary in $")
```

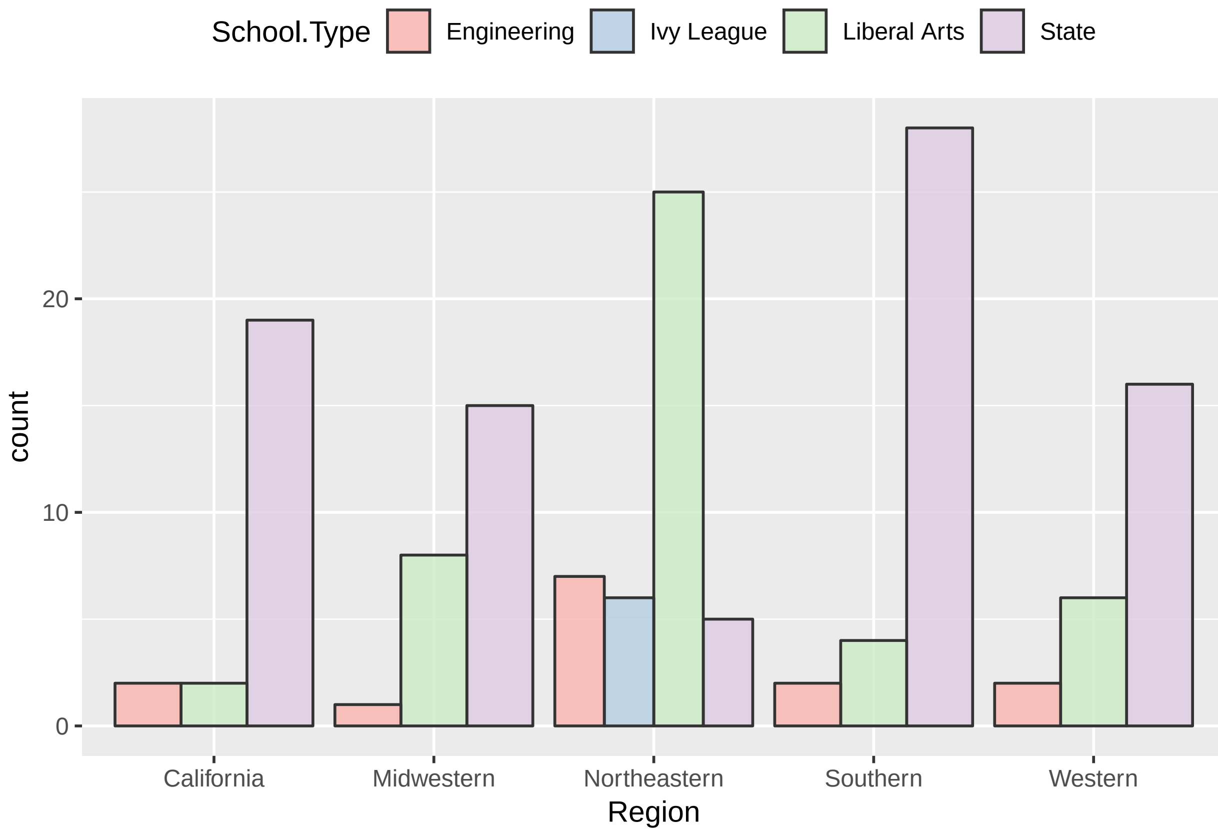
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```




```r
## Count by college type

ggplot(data, aes(School.Type)) +
  geom_bar(color = 'black', alpha = 0.8)
```

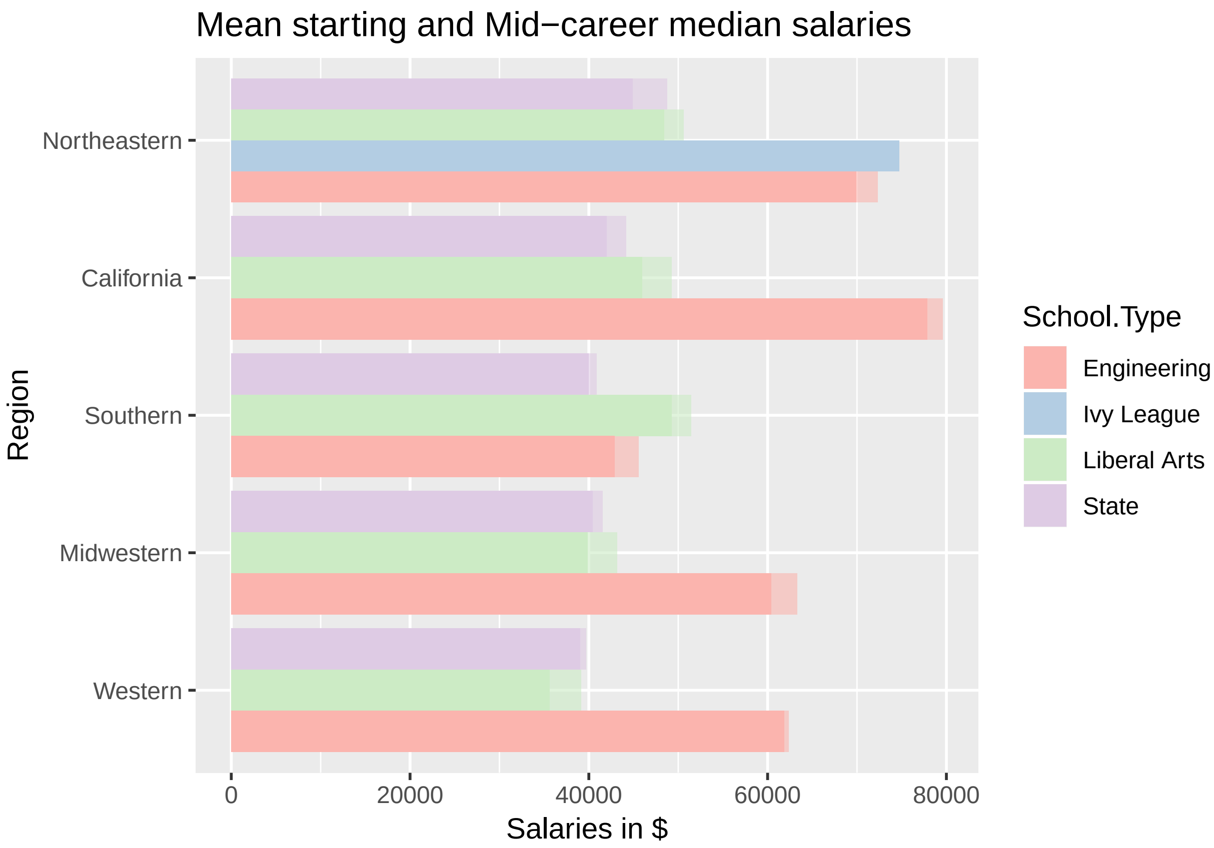



```
##  Region and type
  ggplot(data, aes(Region, fill = School.Type)) +
    geom_bar(position = 'dodge', alpha = 0.8, color = 'gray20') +
    scale_fill_brewer(palette = 'Pastel1') +
    theme(legend.position = "top")
```




```
# How do starting salary and mid-career median salary differ over region and type?

#Below is a look at the mean starting and mid-career salaries over these two categories.
  ggplot(data, aes(reorder(Region, MN_EARN_WNE_P6), MD_EARN_WNE_P8, fill = School.Type)) +
    stat_summary(geom = 'col', position = 'dodge', alpha = 0.6) +
    stat_summary(aes(Region, MN_EARN_WNE_P6, fill = School.Type),
                 geom = 'col', position = 'dodge') +
    scale_fill_brewer(palette = 'Pastel1') +
      xlab('Region') +
    ylab('Salaries in $') +
    ggtitle('Mean starting and Mid-career median salaries') +
    coord_flip()
```

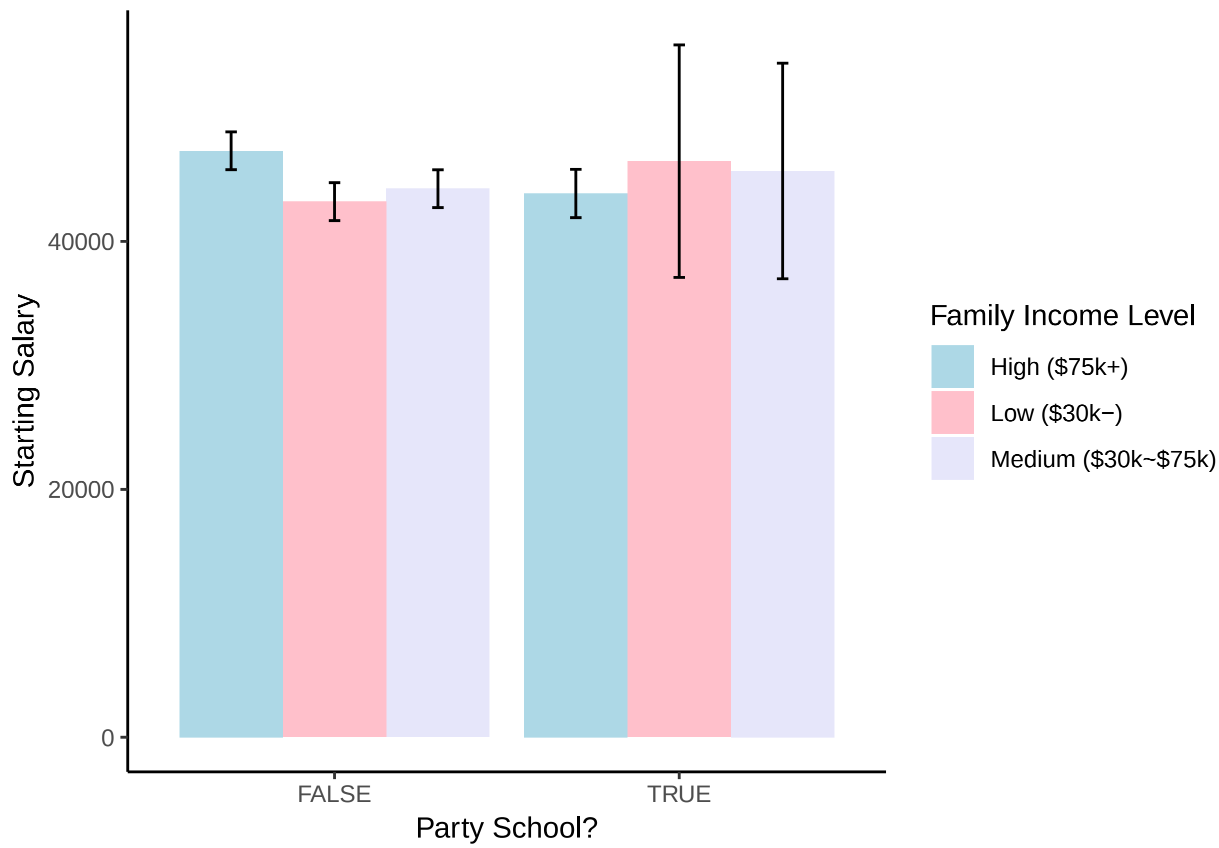
```
## No summary function supplied, defaulting to `mean_se()

## No summary function supplied, defaulting to `mean_se()
```



Gender & Family

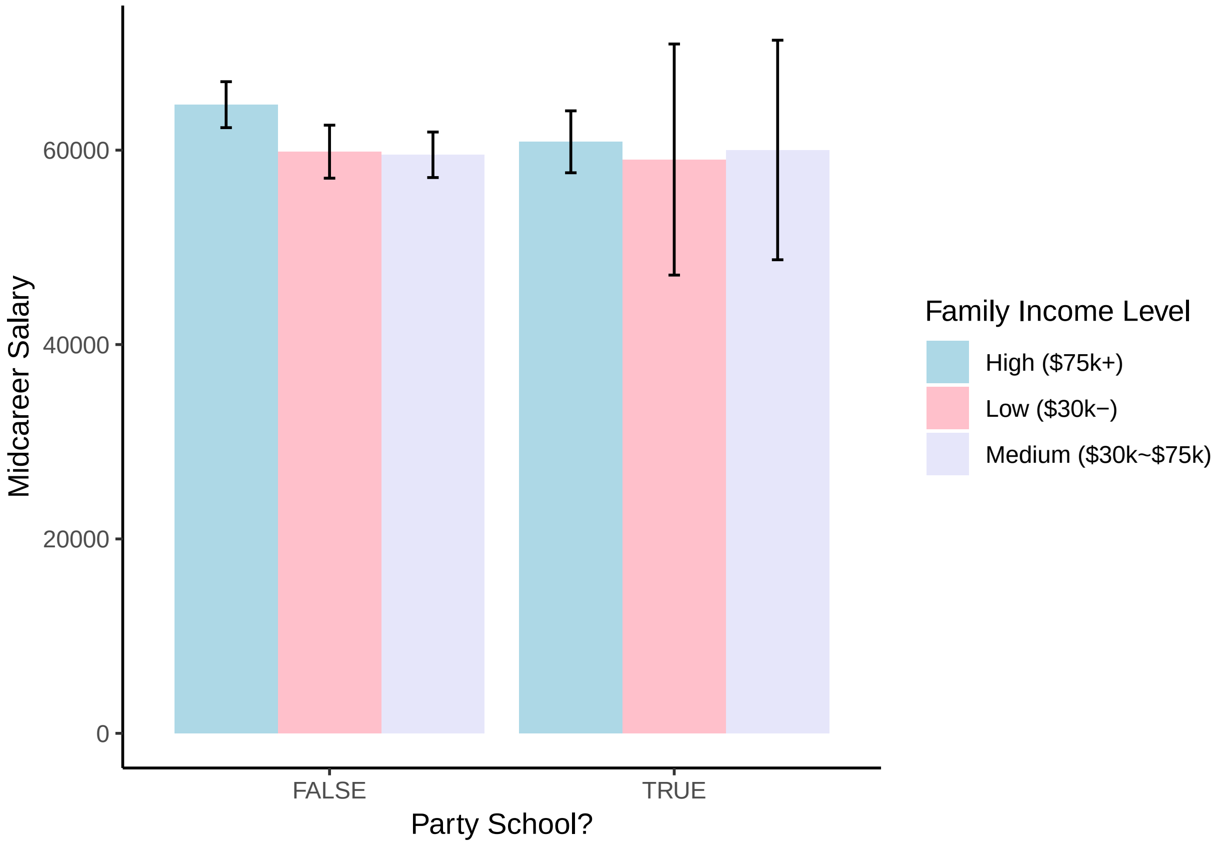
```
#Wrangle data to add gender variable
data_long <- gather(data, "gender", "value", MN_EARN_WNE_MALE0_P6, MN_EARN_WNE_MALE1_P6, MN_EARN_WNE_MAL
data_long1 <- separate(data_long, gender, c("gender", "gendered"), sep = "_P", remove = FALSE)
data_long2 <- spread(data_long1, gendered, value, fill = NA, convert = FALSE, sep = "_Income_P")
data_long2$gender <- str_replace(data_long2$gender, "MN_EARN_WNE_MALE0", "Female")
data_long2$gender <- str_replace(data_long2$gender, "MN_EARN_WNE_MALE1", "Male")

#Wrangle data to add family income variable
data_long3 <- gather(data_long2, "family_income_level", "value", MN_EARN_WNE_INC1_P6, MN_EARN_WNE_INC2_
data_long4 <- separate(data_long3, family_income_level, c("family_income_level", "born"), sep = "_P", r
data_long5 <- spread(data_long4, born, value, fill = NA, convert = FALSE, sep = "_Income_P")
data_long5$family_income_level <- str_replace(data_long5$family_income_level, "MN_EARN_WNE_INC1", "Low"
data_long5$family_income_level <- str_replace(data_long5$family_income_level, "MN_EARN_WNE_INC2", "Medi
data_long5$family_income_level <- str_replace(data_long5$family_income_level, "MN_EARN_WNE_INC3", "High

#Remove NA value
longdata <- drop_na(data_long5)

#Explore the interaction effect with what you can't change - family and gender
longchar =
  ggplot(longdata, aes(longdata$Is.Party,longdata$born_Income_P6,fill=longdata$family_income_level)) +
  stat_summary(fun.y = mean,
```

```
                 geom = "bar",
                 position = "dodge")+
  stat_summary(fun.data = mean_cl_normal,
                 geom = "errorbar",
                 position = position_dodge(width = 0.9),
                 width = 0.1) +
  xlab("Party School?")+
  ylab("Starting Salary") +
  theme_classic() +
  scale_fill_manual(name = "Family Income Level",
                      labels = c("High ($75k+)","Low ($30k-)", "Medium ($30k~$75k)"),
                      values = c("LightBlue","Pink", "Lavender"))
longchar
```

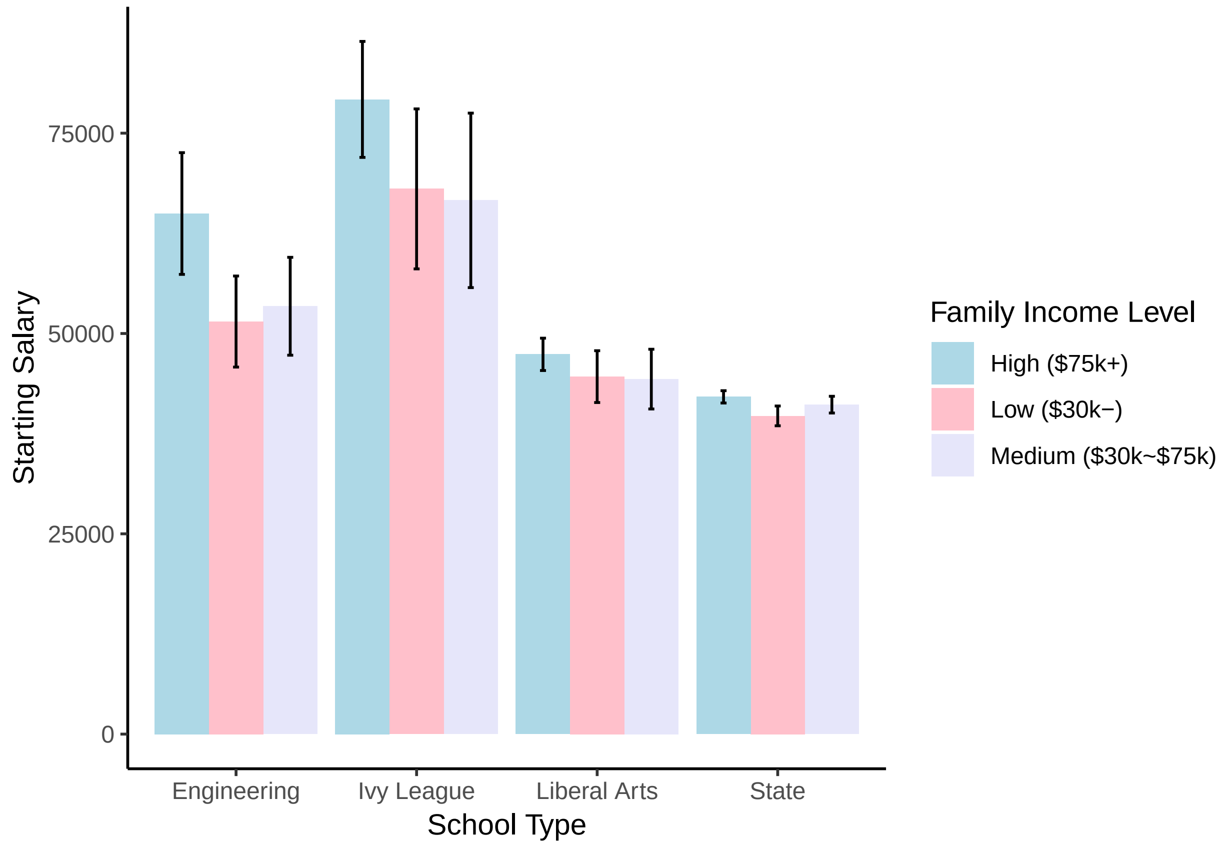



```
longchar2 =
  ggplot(longdata, aes(longdata$Is.Party,longdata$born_Income_P10,fill=longdata$family_income_level)) +
  stat_summary(fun.y = mean,
                 geom = "bar",
                 position = "dodge")+
  stat_summary(fun.data = mean_cl_normal,
                 geom = "errorbar",
                 position = position_dodge(width = 0.9),
                 width = 0.1) +
  xlab("Party School?")+
  ylab("Midcareer Salary") +
  theme_classic() +
  scale_fill_manual(name = "Family Income Level",
```

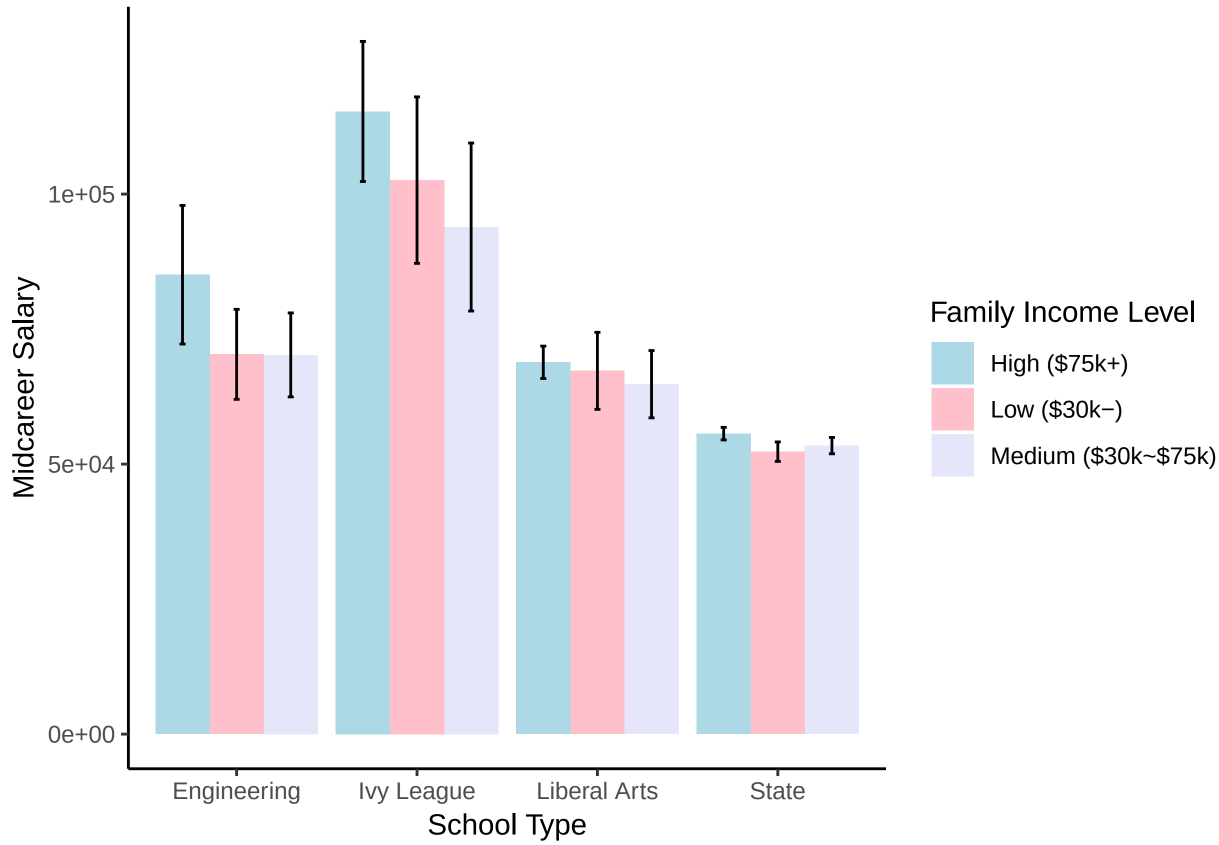
```
                    labels = c("High ($75k+)","Low ($30k-)", "Medium ($30k~$75k)"),
                    values = c("LightBlue","Pink", "Lavender"))
longchar2
```



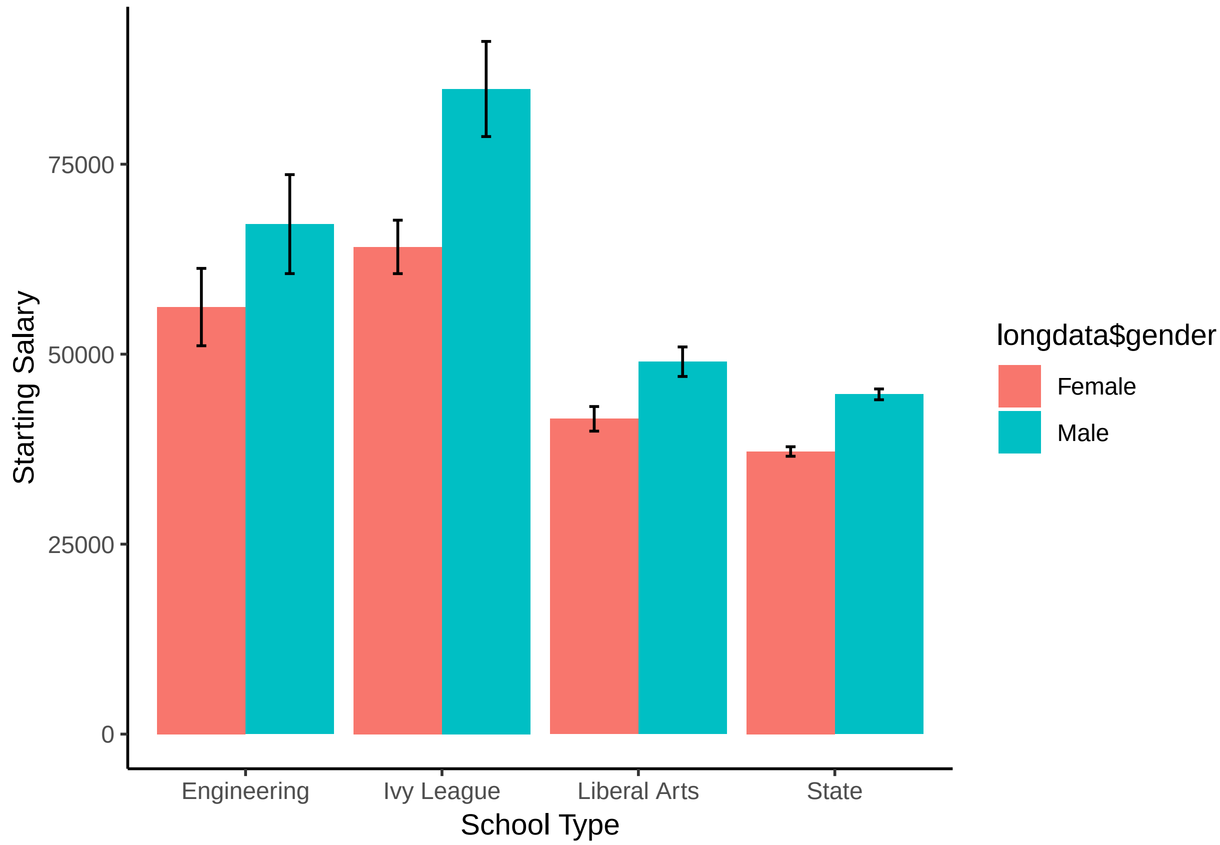

```
longchar3 =
  ggplot(longdata, aes(longdata$School.Type,longdata$born_Income_P6,fill=longdata$family_income_level))
  stat_summary(fun.y = mean,
             geom = "bar",
             position = "dodge")+
  stat_summary(fun.data = mean_cl_normal,
             geom = "errorbar",
             position = position_dodge(width = 0.9),
             width = 0.1) +
  xlab("School Type")+
  ylab("Starting Salary") +
  theme_classic() +
  scale_fill_manual(name = "Family Income Level",
                    labels = c("High ($75k+)","Low ($30k-)", "Medium ($30k~$75k)"),
                    values = c("LightBlue","Pink", "Lavender"))
longchar3
```



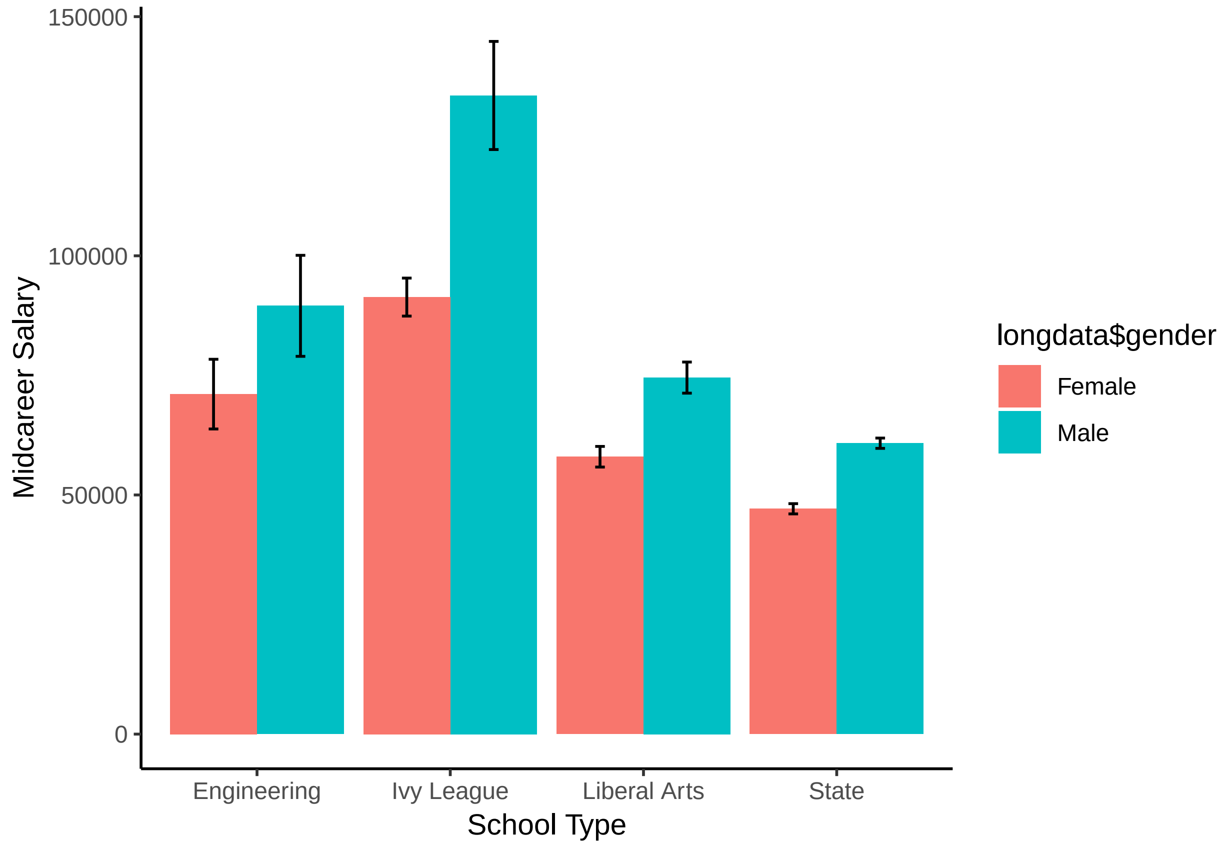

```
longchar4 =
  ggplot(longdata, aes(longdata$School.Type,longdata$born_Income_P10,fill=longdata$family_income_level))
  stat_summary(fun.y = mean,
               geom = "bar",
               position = "dodge")+
  stat_summary(fun.data = mean_cl_normal,
               geom = "errorbar",
               position = position_dodge(width = 0.9),
               width = 0.1) +
  xlab("School Type")+
  ylab("Midcareer Salary") +
  theme_classic() +
  scale_fill_manual(name = "Family Income Level",
                    labels = c("High ($75k+)","Low ($30k-)", "Medium ($30k~$75k)"),
                    values = c("LightBlue","Pink", "Lavender"))
longchar4
```




```
longchar5 =
  ggplot(longdata, aes(longdata$School.Type,longdata$gendered_Income_P6,fill=longdata$gender)) +
  stat_summary(fun.y = mean,
               geom = "bar",
               position = "dodge")+
  stat_summary(fun.data = mean_cl_normal,
               geom = "errorbar",
               position = position_dodge(width = 0.9),
               width = 0.1) +
  xlab("School Type")+
  ylab("Starting Salary") +
  theme_classic()
longchar5
```



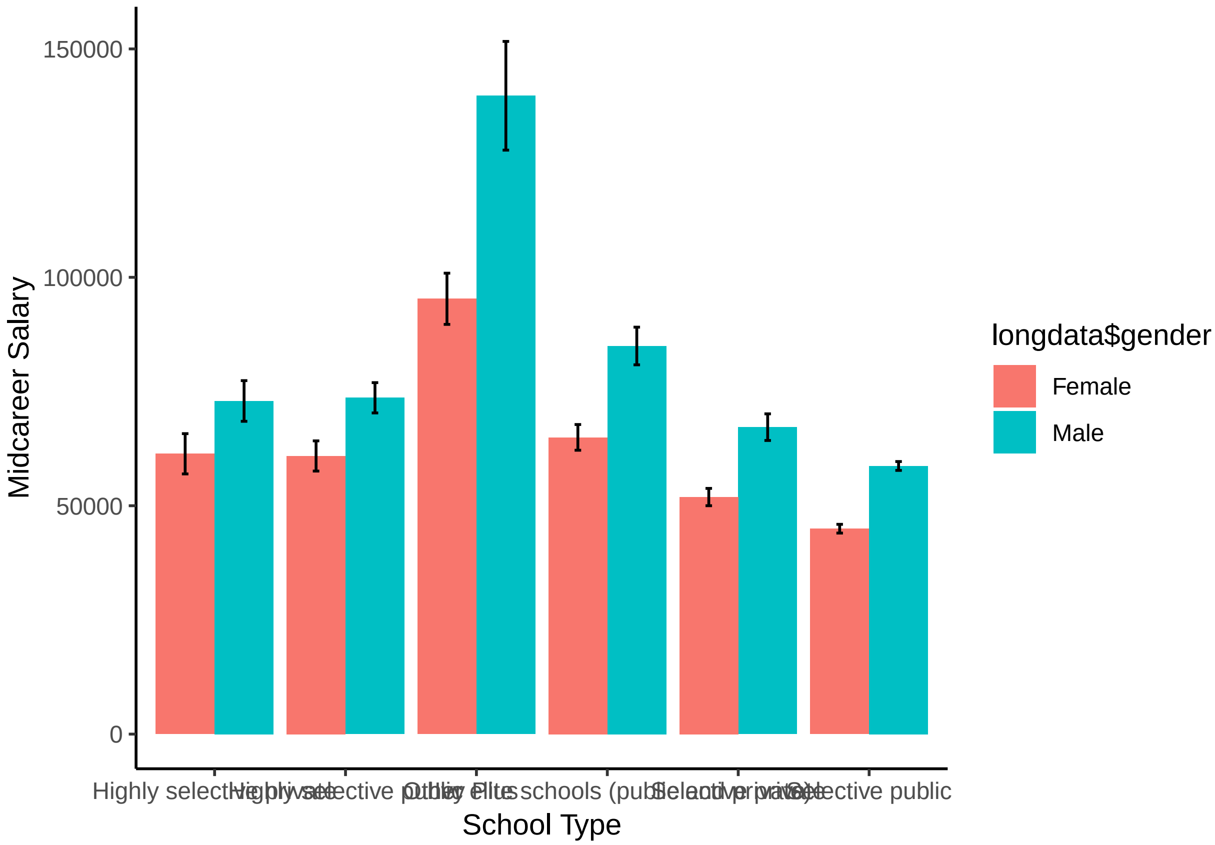

```
longchar6 =
  ggplot(longdata, aes(longdata$School.Type,longdata$gendered_Income_P10,fill=longdata$gender)) +
  stat_summary(fun.y = mean,
               geom = "bar",
               position = "dodge")+
  stat_summary(fun.data = mean_cl_normal,
               geom = "errorbar",
               position = position_dodge(width = 0.9),
               width = 0.1) +
  xlab("School Type")+
  ylab("Midcareer Salary") +
  theme_classic()
longchar6
```




```
unique(longdata$Tier)
```

```
## [1] Selective public
## [2] Other elite schools (public and private)
## [3] Highly selective public
## [4] Highly selective private
## [5] Ivy Plus
## [6] Selective private
## 12 Levels: Four-year for-profit ... Two-year for-profit
```

```
longchar7 =
  ggplot(longdata, aes(longdata$Tier,longdata$gendered_Income_P10,fill=longdata$gender)) +
  stat_summary(fun.y = mean,
               geom = "bar",
               position = "dodge")+
  stat_summary(fun.data = mean_cl_normal,
               geom = "errorbar",
               position = position_dodge(width = 0.9),
               width = 0.1) +
  xlab("School Type")+
  ylab("Midcareer Salary") +
  theme_classic()
longchar7
```




```r
#T-test to see if there is a difference in starting income from the two different gender group
#H1: M1 <> M2
#H0: M1 = M2
t.test(longdata$gendered_Income_P6 ~ longdata$gender)
```

```
##
##  Welch Two Sample t-test
##
## data:  longdata$gendered_Income_P6 by longdata$gender
## t = -9.664, df = 712.38, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -10187.667  -6747.241
## sample estimates:
## mean in group Female   mean in group Male
##             41298.95             49766.40
```

```r
#Power and effect size
table(longdata$gender)
```

```
##
## Female   Male
##    381    381
```

```r
sd(longdata$gendered_Income_P6[longdata$gender=="Female"])
```

```
## [1] 10413.3
```

```r
sd(longdata$gendered_Income_P6[longdata$gender=="Male"])
```

```
## [1] 13566.74
```

```r
library(MOTE)
```

```
## Registered S3 methods overwritten by 'car':
##   method                          from
##   influence.merMod                lme4
##   cooks.distance.influence.merMod lme4
##   dfbeta.influence.merMod         lme4
##   dfbetas.influence.merMod        lme4
```

```r
effect = d.ind.t(m1 = 41038.38, m2 = 49424.02,
                 sd1 = 10289.34, sd2 = 13488.84,
                 n1 = 383, n2 = 383, a = .05)
effect$d
```

```
## [1] -0.6990226
```

```r
library(pwr)
pwr.t.test(n = NULL, d = effect$d,
           sig.level = .05,
           power = .80, type = "two.sample",
           alternative = "two.sided")
```

```
##
##      Two-sample t test power calculation
##
##               n = 33.11414
##               d = 0.6990226
##       sig.level = 0.05
##           power = 0.8
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```