HR Analytics

Identifying the best recruiting source

```
# Load the readr package
library(readr)
```

```
## Warning: package 'readr' was built under R version 3.4.4
```

```
# Import the recruitment data
recruitment <- read_csv("recruitment_data.csv")
```

```
## Parsed with column specification:
## cols(
##   attrition = col_double(),
##   performance_rating = col_double(),
##   sales_quota_pct = col_double(),
##   recruiting_source = col_character()
## )
```

```
# Look at the first few rows of the dataset
head(recruitment)
```

attrition <dbl>	performance_rating <dbl>	sales_quota_pct <dbl>	recruiting_source <chr>
1	3	1.0881902	Applied Online
0	3	2.3941726	NA
1	2	0.4975302	Campus
0	2	2.5139577	NA
0	3	1.4247888	Applied Online
1	3	0.5481232	Referral

6 rows

```
# Load the dplyr package
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.4.4
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
# Get an overview of the recruitment data
summary(recruitment)
```

```
##    attrition     performance_rating sales_quota_pct   recruiting_source
##  Min.   :0.000   Min.   :1.000      Min.   :-0.7108   Length:446
##  1st Qu.:0.000   1st Qu.:2.000      1st Qu.: 0.5844   Class :character
##  Median :0.000   Median :3.000      Median : 1.0701   Mode  :character
##  Mean   :0.213   Mean   :2.895      Mean   : 1.0826
##  3rd Qu.:0.000   3rd Qu.:3.000      3rd Qu.: 1.5325
##  Max.   :1.000   Max.   :5.000      Max.   : 3.6667
```

```
# See which recruiting sources the company has been using
count(recruitment, recruiting_source)
```

recruiting_source <chr>	n <int>
Applied Online	130
Campus	56
Referral	45
Search Firm	10
NA	205

5 rows

Which recruiting channel produces the best salespeople? One quality of hire metric you can use is sales quota attainment, or how much a salesperson sold last year relative to their quota. An employee whose sales_quota_pct equals .75 sold 75% of their quota, for example. This metric can be helpful because raw sales numbers are not always comparable between employees.

We will Calculate the average sales quota attainment achieved by hires from each recruiting source.

```
# Find the average sales quota attainment
recruitment %>%
  summarize(avg_sales_quota_pct = mean(sales_quota_pct, na.rm = TRUE))
```

avg_sales_quota_pct <dbl>
1.082607

1 row

Use summarize() to calculate the average sales quota attainment within each recruiting source. Store it in a new column called avg_sales_quota_pct. Assign the result to avg_sales.

```
# Find the average sales quota attainment for each recruiting source
avg_sales <- recruitment %>%
      group_by(recruiting_source) %>%
  summarize(avg_sales_quota_pct = mean(sales_quota_pct, na.rm = TRUE))

# Display the result
avg_sales
```

recruiting_source <chr>	avg_sales_quota_pct <dbl>
Applied Online	1.0585902
Campus	0.9080354
Referral	1.0231982
Search Firm	0.8869603
NA	1.1681091

5 rows

Another quality of hire metric you can consider is the attrition rate, or how often hires leave the company. Determine which recruiting channels have the highest and lowest attrition rates.

```
# Find the average attrition for the sales team, by recruiting source, sorted from lowest att
rition rate to highest
avg_attrition <- recruitment %>%
  group_by(recruiting_source) %>%
  summarize(attrition_rate = mean(attrition, na.rm = TRUE))%>%
  arrange(avg_attrition = attrition_rate)
```

```
## Warning: package 'bindrcpp' was built under R version 3.4.4
```

```
# Display the result
avg_attrition
```

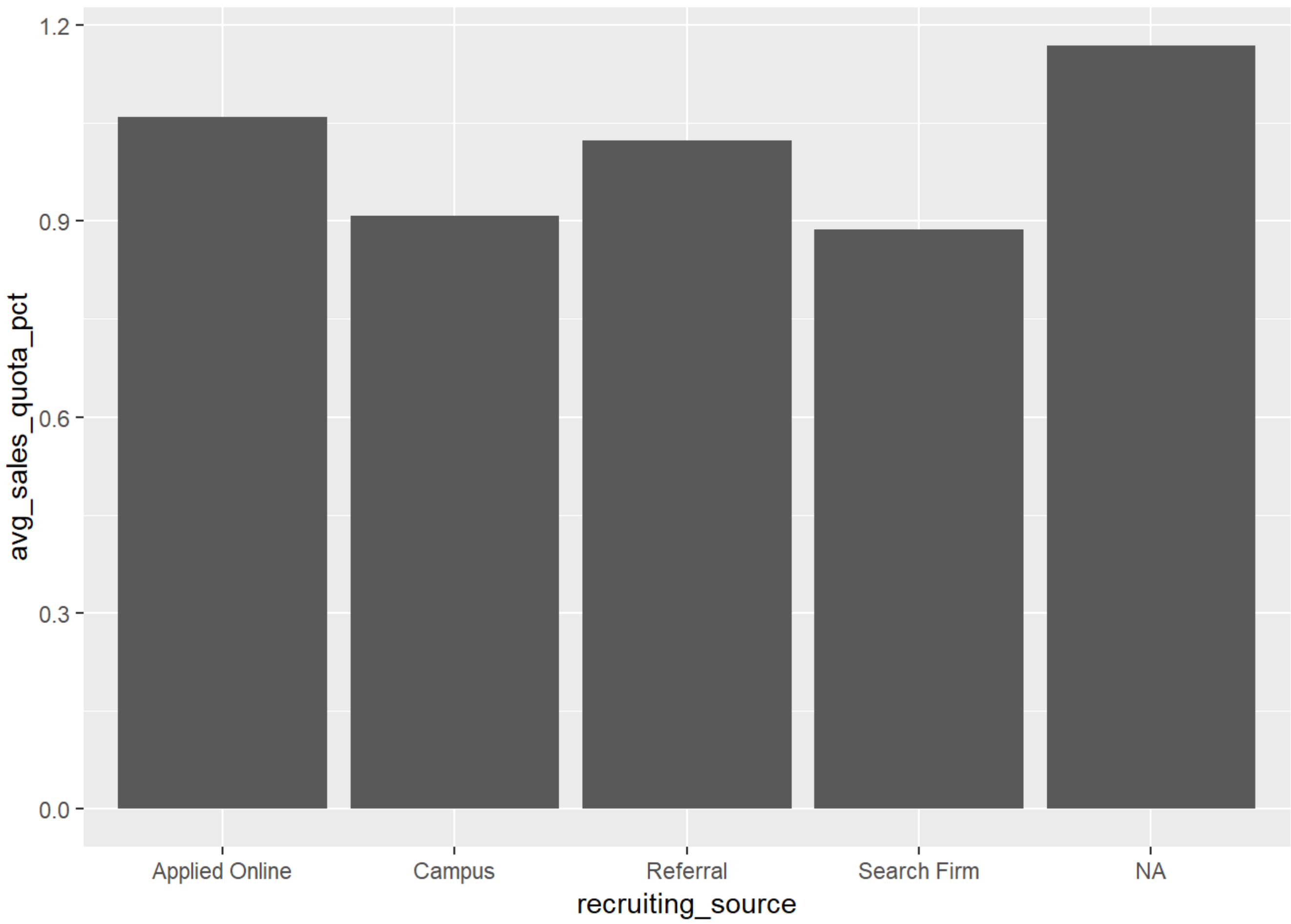
recruiting_source <chr>	attrition_rate <dbl>
NA	0.1317073
Applied Online	0.2461538
Campus	0.2857143
Referral	0.3333333
Search Firm	0.5000000

5 rows

```
# Load the ggplot2 package
library(ggplot2)
```

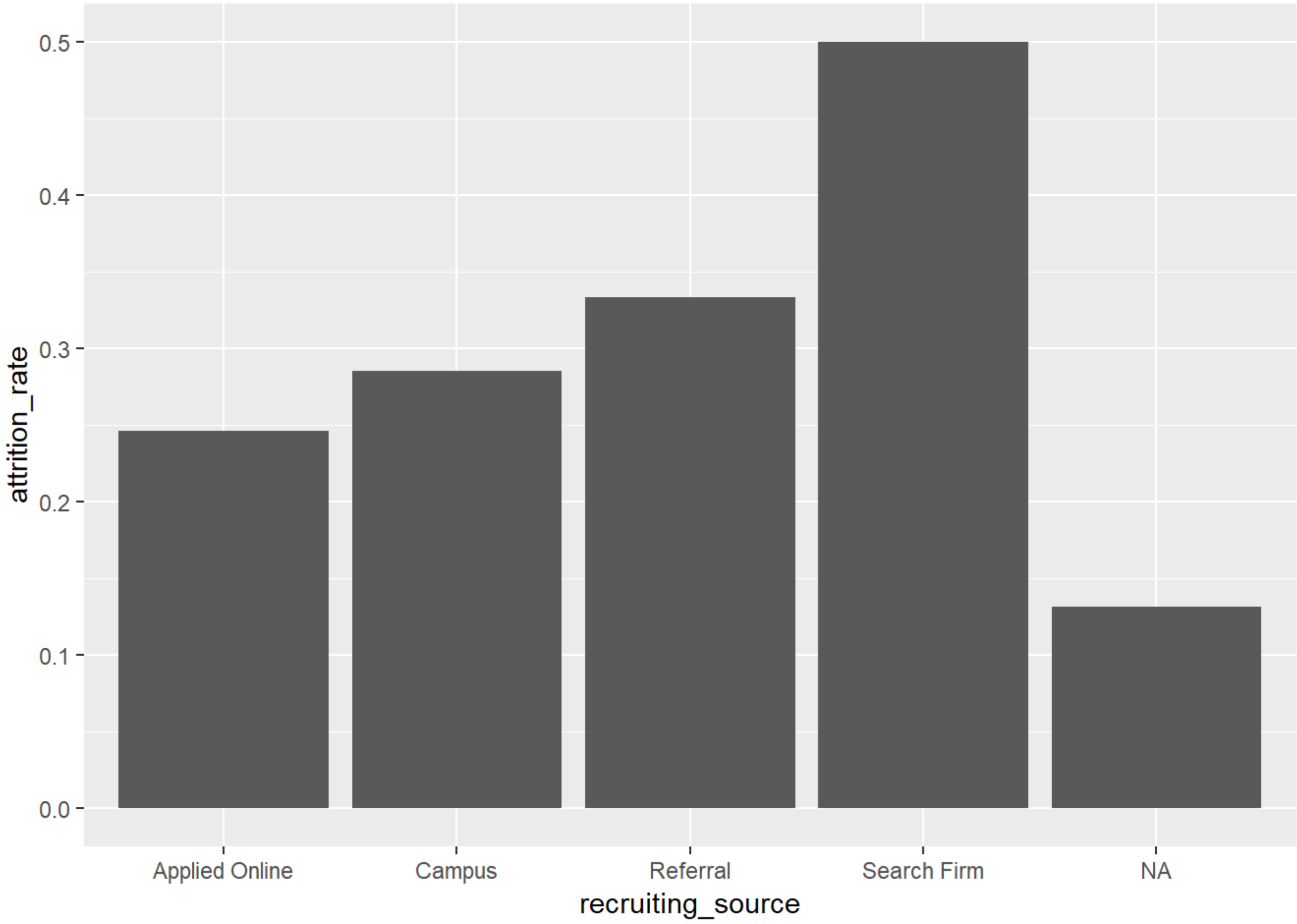
```
## Warning: package 'ggplot2' was built under R version 3.4.4
```

```
# Plot the bar chart
ggplot(avg_sales, aes(x=recruiting_source, y=avg_sales_quota_pct)) + geom_col()
```



Attrition Rates. Bar chart of avergae attrition

```
ggplot(avg_attrition, aes(x=recruiting_source, y=attrition_rate)) + geom_col()
```



Conclusion: You cannnot say NA is best, as NA indicates the hiring source is missing. The best source is Applied Online and the worst source is Search Firm.

What is driving low employee engagement?

```
survey <- read_csv("survey_data.csv")
```

```
## Parsed with column specification:
## cols(
##   employee_id = col_double(),
##   department = col_character(),
##   engagement = col_double(),
##   salary = col_double(),
##   vacation_days_taken = col_double()
## )
```

```
summary(survey)
```

```
##   employee_id      department          engagement       salary
##  Min.   :   1.0   Length:1470        Min.   :1.00   Min.   : 45530
##  1st Qu.: 491.2   Class :character   1st Qu.:3.00   1st Qu.: 59407
##  Median :1020.5   Mode  :character   Median :3.00   Median : 70481
##  Mean   :1024.9                      Mean   :3.05   Mean   : 74162
##  3rd Qu.:1555.8                      3rd Qu.:4.00   3rd Qu.: 84763
##  Max.   :2068.0                      Max.   :5.00   Max.   :164073
##  vacation_days_taken
##  Min.   : 0.00
##  1st Qu.: 6.00
##  Median :10.00
##  Mean   :11.27
##  3rd Qu.:16.00
##  Max.   :38.00
```

```
#Use count() on the department variable, since summary() doesn't provide much information abo
ut character variables.
count(survey, department)
```

department <chr>	n <int>
Engineering	961
Finance	63
Sales	446

3 rows

Which department has the lowest engagement?

```
survey %>%
  group_by(department) %>%
  summarize(avg_engagement = mean(engagement)) %>%
  arrange(avg_engagement)
```

department <chr>	avg_engagement <dbl>
Sales	2.807175
Engineering	3.150884
Finance	3.238095

3 rows

Another common way to think about engagement is identifying which employees are disengaged, which we'll define as having an engagement score of 1 or 2. The survey dataset doesn't have a column called disengaged, but we will create it.

```
survey_disengaged <- survey %>%
  mutate(disengaged = ifelse(engagement <= 2, 1, 0))

survey_disengaged
```

employee_id <dbl>	department <chr>	engagement <dbl>	salary <dbl>	vacation_days_taken <dbl>	disengaged <dbl>
1	Sales	3	103263.64	7	0
2	Engineering	3	80708.64	12	0
4	Engineering	3	60737.05	12	0
5	Engineering	3	99116.32	7	0
7	Engineering	3	51021.64	18	0
8	Engineering	3	98399.87	9	0
10	Engineering	3	57106.20	18	0
11	Engineering	1	55065.03	4	1
12	Engineering	4	77158.03	12	0
13	Engineering	2	48364.62	14	1

1-10 of 1,470 rows Previous **1** 2 3 4 5 6 ... 147 Next

```
survey_summary <- survey_disengaged %>%
  group_by(department) %>%
  summarize(pct_disengaged = mean(disengaged),
            avg_salary = mean(salary),
            avg_vacation_days = mean(vacation_days_taken))

survey_summary
```

department <chr>	pct_disengaged <dbl>	avg_salary <dbl>	avg_vacation_days <dbl>
Engineering	0.2060354	73576.35	12.204995
Finance	0.1904762	76651.66	11.476190
Sales	0.3295964	75073.57	9.224215

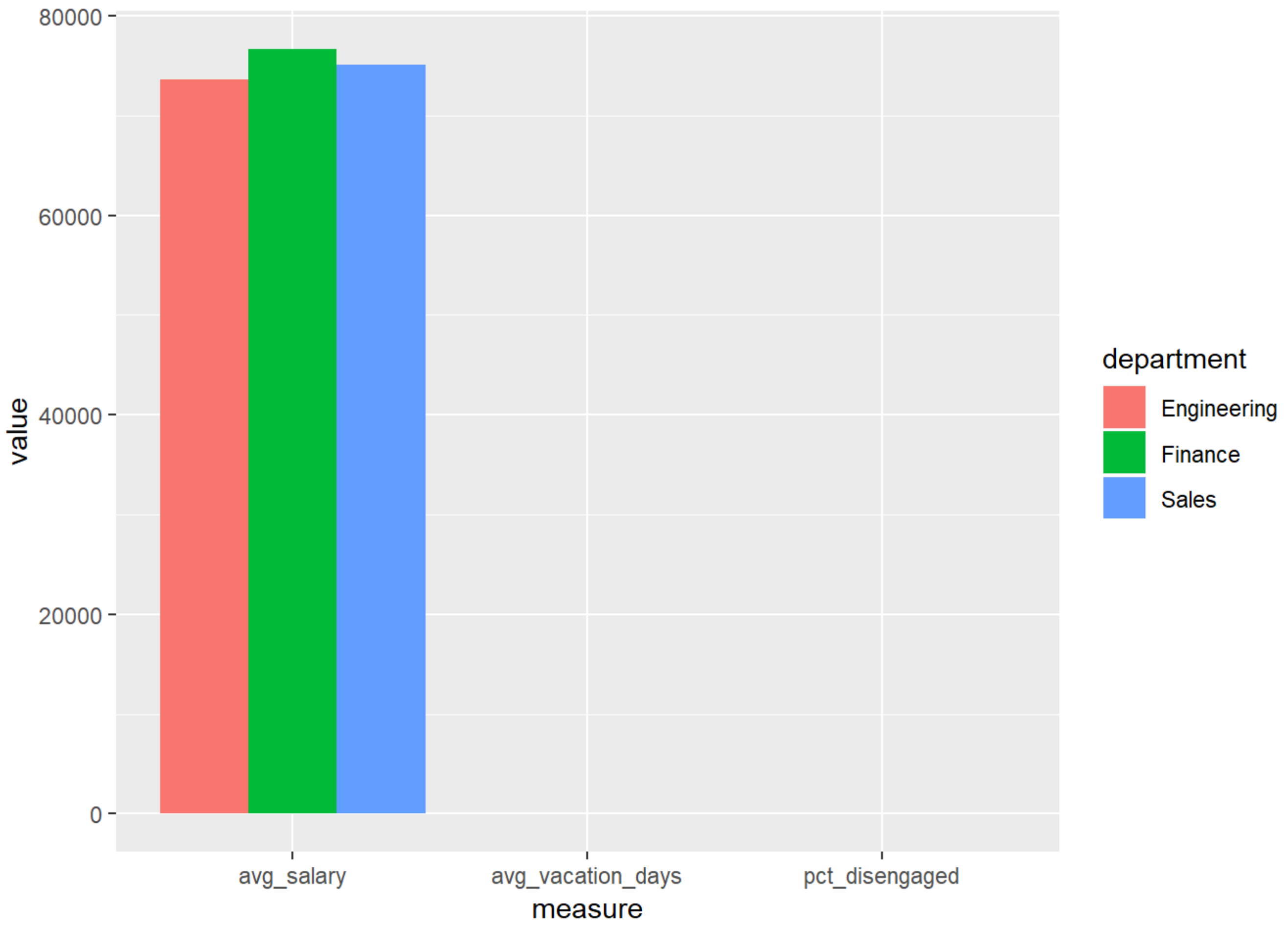
3 rows

```
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 3.4.4
```

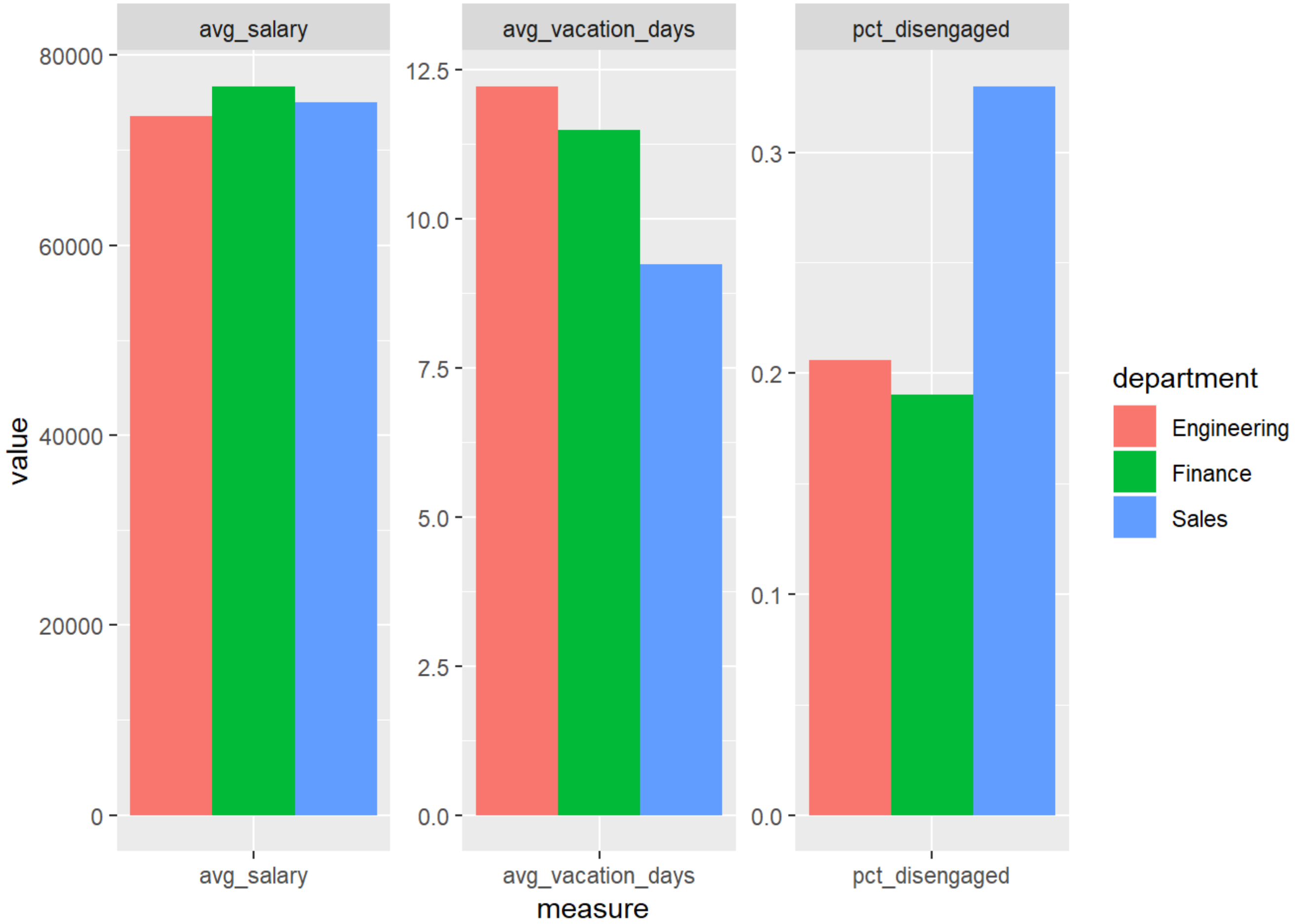
```
survey_gathered <- survey_summary %>%
  gather(key = "measure", value = "value",
         pct_disengaged, avg_salary, avg_vacation_days)

# Create three bar charts
ggplot(survey_gathered, aes(measure, value, fill = department)) +
  geom_col(position = "dodge")
```



Two of the bar charts are very tiny.

```
# Create three faceted bar charts
ggplot(survey_gathered, aes(measure, value, fill=department))+ geom_col(position='dodge') + f
acet_wrap(facet= ~measure, scales="free")
```




```
survey_disengaged <- survey %>%
  mutate(disengaged = ifelse(engagement <= 2, 1, 0))
```

Some inference we could draw from the graph: Sales department has the highest disengaged employees and it also has the least vacation days. We now need to check if the difference is statistically significant.

We've seen some evidence that the sales department has a higher proportion of disengaged employees than the rest of the company, but we aren't yet certain if that difference is significant. We can test whether that difference is statistically significant using the chi-squared test. Chi-squared is used for categorical features. The t-test is used for continuous features.

```
#Add the in_sales variable, which should be "Sales" for employees in the sales department, an
d "Other" otherwise. Assign the result to survey_sales.
survey_sales <- survey_disengaged %>%
  mutate(in_sales = ifelse(department=='Sales', "Sales", "Other"))

#Use the chi-square the test to test the hypothesis that the sales department has the same pr
oportion of disengaged employees as the rest of the company.
chisq.test(survey_sales$in_sales, survey_sales$disengaged)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  survey_sales$in_sales and survey_sales$disengaged
## X-squared = 25.524, df = 1, p-value = 4.368e-07
```

```
# Is the result significant? Yes since the p-value is less than 0.05
significant <- TRUE
```

The other observation was that employees in the sales department take fewer vacation days on average than the rest of the company. We can test whether that observation is statistically significant as well.

```
t.test(vacation_days_taken ~ in_sales, data = survey_sales)
```

```
## 
##  Welch Two Sample t-test
## 
## data:  vacation_days_taken by in_sales
## t = 8.1549, df = 1022.9, p-value = 1.016e-15
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  2.229473 3.642409
## sample estimates:
## mean in group Other mean in group Sales 
##           12.160156            9.224215
```

Since the p is less than 0.05 then the test is statistically significant

Are new hires getting paid too much?

When employers make a new hire, they must determine what the new employee will be paid. If the employer is not careful, the new hires can come in with a higher salary than the employees that currently work at the same job, which can cause employee turnover and dissatisfaction. In this chapter, you will check whether new hires are really getting paid more than current employees, and how to double-check your initial observations.

```
# Import the data
pay <- read_csv('fair_pay_data.csv')
```

```
## Parsed with column specification:
## cols(
##   employee_id = col_double(),
##   department = col_character(),
##   salary = col_double(),
##   new_hire = col_character(),
##   job_level = col_character()
## )
```

```
summary(pay)
```

```
##   employee_id      department           salary        new_hire
##  Min.   :   1.0   Length:1470        Min.   : 43820   Length:1470
##  1st Qu.: 491.2   Class :character   1st Qu.: 59378   Class :character
##  Median :1020.5   Mode  :character   Median : 70425   Mode  :character
##  Mean   :1024.9                      Mean   : 74142
##  3rd Qu.:1555.8                      3rd Qu.: 84809
##  Max.   :2068.0                      Max.   :164073
##   job_level
##  Length:1470
##  Class :character
##  Mode  :character
##
##
##
```

```
# Check average salary of new hires and non-new hires
pay %>%
  group_by(new_hire) %>%
    summarize(avg_salary = mean(salary))
```

new_hire <chr>	avg_salary <dbl>
No	73424.60
Yes	76074.28

2 rows

It looks like new hires are being paid more than current employees. We will now check if the differnce is statistically significant.

```
t.test(salary ~ new_hire, data = pay)
```

```
##
##  Welch Two Sample t-test
##
## data:  salary by new_hire
## t = -2.3437, df = 685.16, p-value = 0.01938
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -4869.4242  -429.9199
## sample estimates:
##  mean in group No mean in group Yes
##          73424.60          76074.28
```

```
# Do the same test, and tidy up the output
library(broom)
```

```
## Warning: package 'broom' was built under R version 3.4.4
```

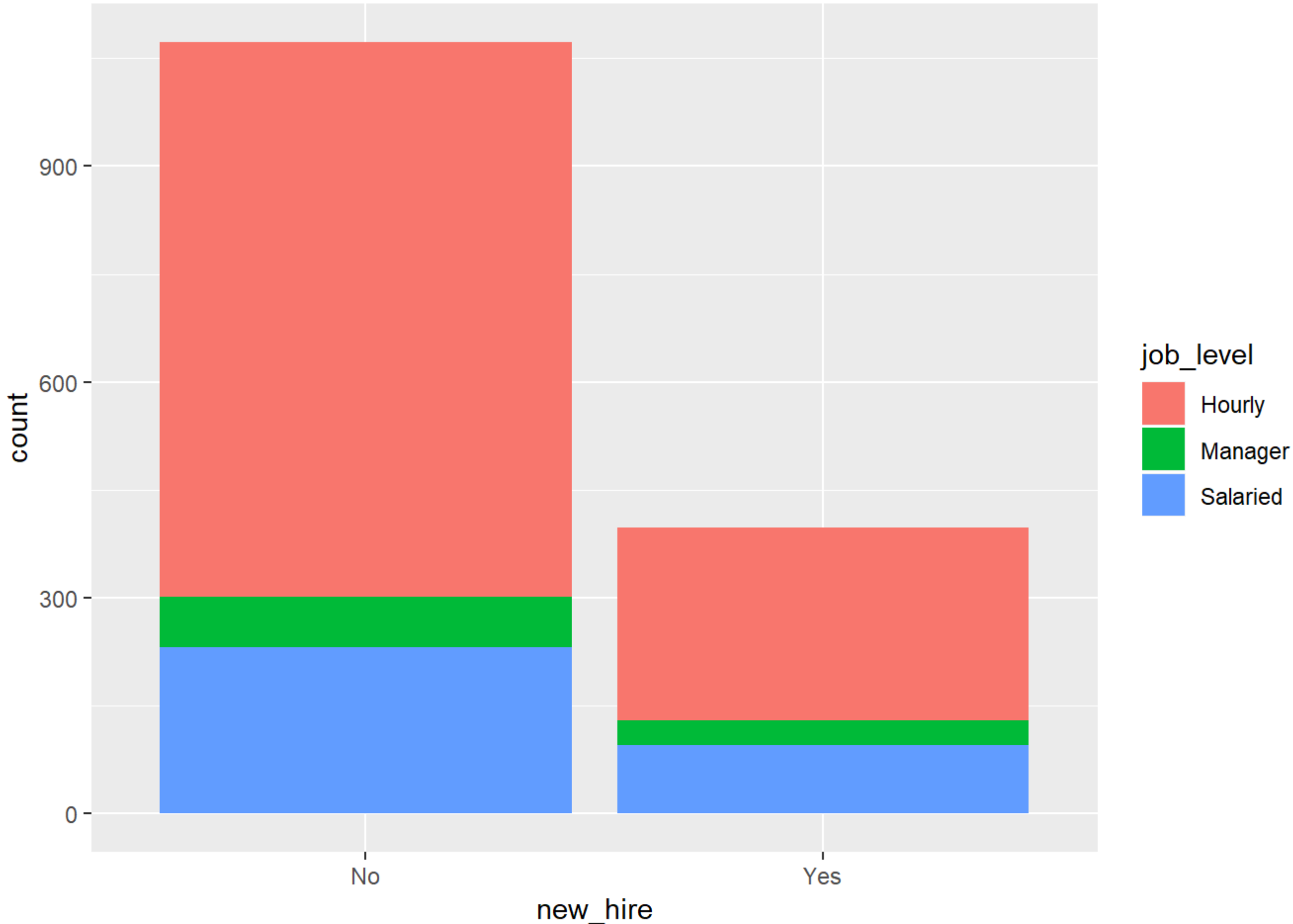
```
t.test(salary ~ new_hire, data = pay) %>%
  tidy()
```

estimate <dbl>	estimate1 <dbl>	estimate2 <dbl>	statistic <dbl>	p.value <dbl>	parameter <dbl>	conf.low <dbl>	conf.high <dbl>
-2649.672	73424.6	76074.28	-2.343708	0.01937799	685.1554	-4869.424	-429.9199

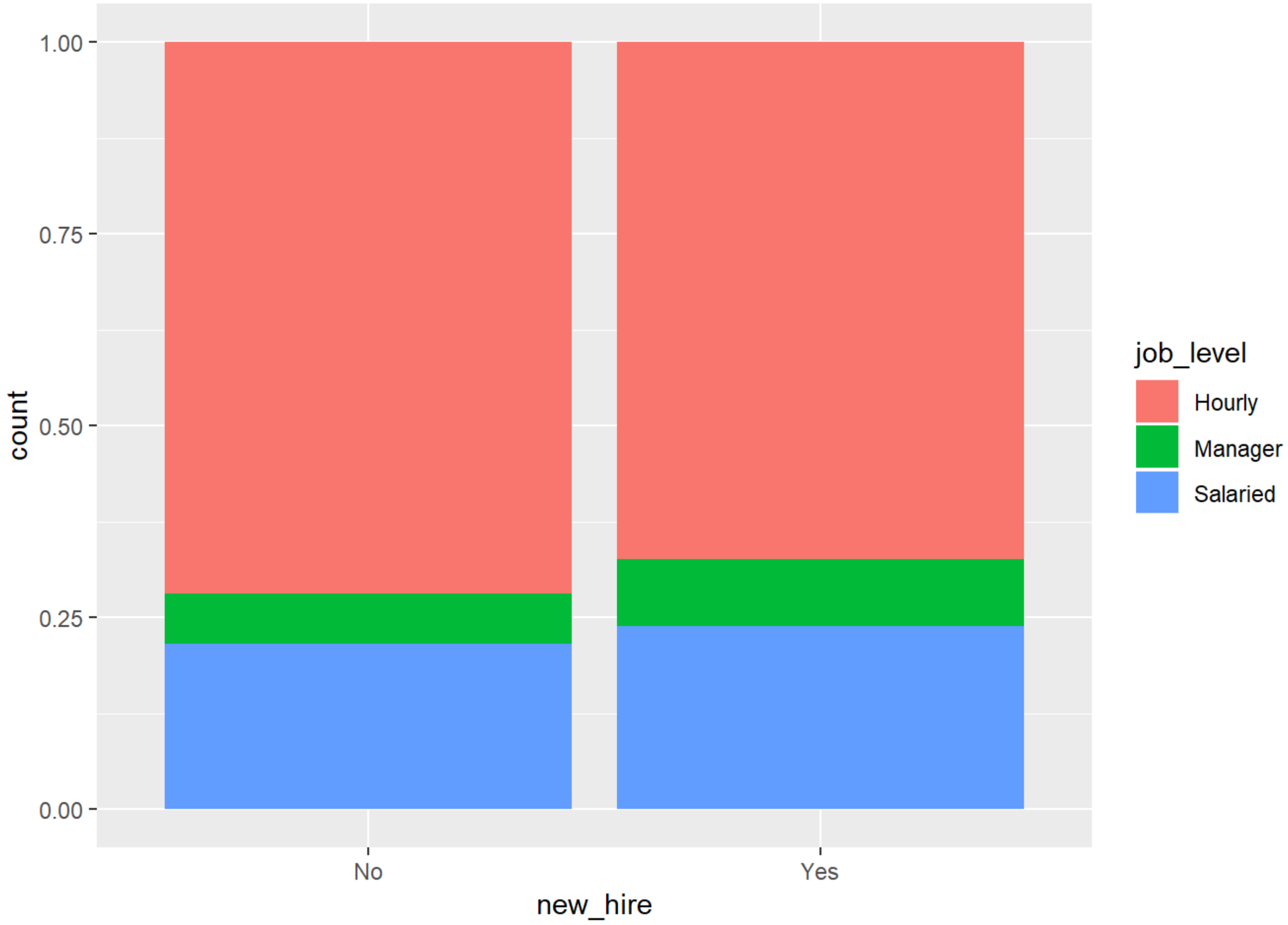
1 row | 1-8 of 10 columns

From the p-value we can see that there is a significal difference in the salary.

```
# Create a stacked bar chart
pay %>%
  ggplot(aes(x=new_hire, fill=job_level)) + geom_bar()
```




```
pay %>%
  ggplot(aes(x=new_hire, fill=job_level)) + geom_bar(position='fill')
```

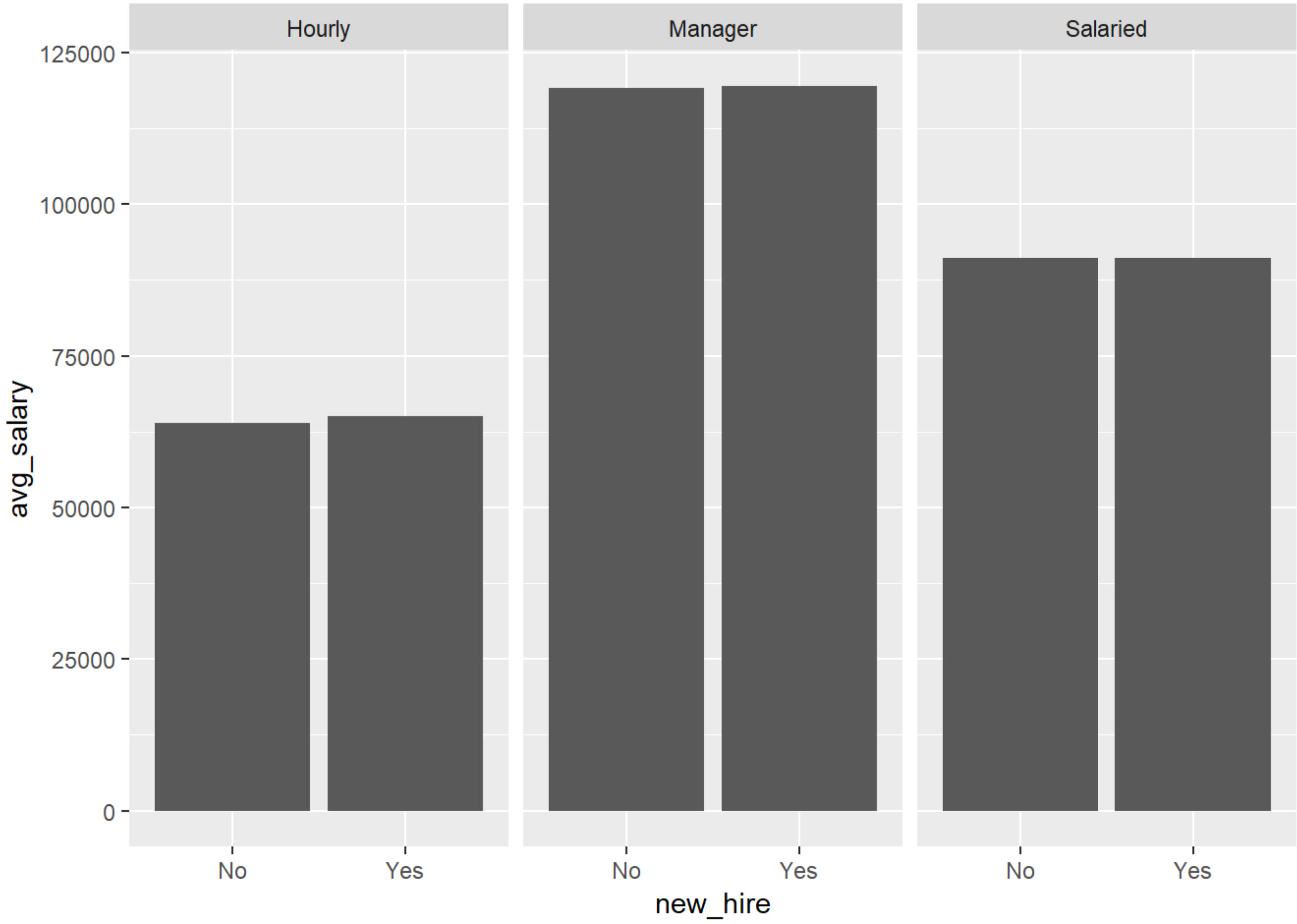


It looks like new hires are less likely to be hourly employees than current employees.

Do new hires have a higher average salary than current employees when job level is taken into account? Calculate the average salaries, and then recreate the bar chart from earlier in the chapter, adding faceting to split it up by the three job levels. Are the bar heights closer together than they were in the first plot?

```
# Calculate the average salary for each group of interest
pay_grouped <- pay %>%
  group_by(new_hire, job_level) %>%
  summarize(avg_salary = mean(salary))

# Graph the results using facet_wrap()
pay_grouped %>%
  ggplot(aes(x=new_hire, y=avg_salary))+geom_col()+facet_wrap(facets=~job_level)
```



In the plot you made, the bars were nearly equal. This supports the idea that an omitted variable - job level - is driving the difference in pay for new hires and current employees. However, the graph shows a small difference in the average salaries for hourly workers. Test whether a significant pay difference exists between hourly new hires and hourly current employees.

```
pay_filter <- pay %>%
  filter(job_level=='Hourly')

t.test(salary ~ new_hire, data = pay_filter) %>%
  tidy()
```

estimate <dbl>	estimate1 <dbl>	estimate2 <dbl>	statistic <dbl>	p.value <dbl>	parameter <dbl>	conf.low <dbl>	conf.high <dbl>
-1106.967	63965.71	65072.68	-1.750387	0.08066517	499.7005	-2349.483	135.5483

1 row | 1-8 of 10 columns

The difference is not statistically significant

```
# Run the simple regression
model_simple <- lm(salary ~ new_hire, data = pay)

# Display the summary of model_simple
model_simple %>%
  summary()
```

```
##
## Call:
## lm(formula = salary ~ new_hire, data = pay)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -32255 -14466  -3681  10740  87998
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  73424.6      577.2 127.200   <2e-16 ***
## new_hireYes   2649.7     1109.4   2.388    0.017 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18900 on 1468 degrees of freedom
## Multiple R-squared:  0.003871,   Adjusted R-squared:  0.003193
## F-statistic: 5.705 on 1 and 1468 DF,  p-value: 0.01704
```

```
# Display a tidy summary
model_simple %>%
  tidy()
```

term <chr>	estimate <dbl>	std.error <dbl>	statistic <dbl>	p.value <dbl>
(Intercept)	73424.603	577.2369	127.200112	0.00000000
new_hireYes	2649.672	1109.3568	2.388476	0.01704414

2 rows